NOIDA INSTITUTE OF ENGINEERING AND TECHNOLOGY, GREATER NOIDA, GAUTAM BUDDH NAGAR

(An Autonomous Institute)

NIET
Greater Noida
An Autonomous Institute
Estd.2001
Get Future Ready

Affiliated to

DR. A.P.J. ABDUL KALAM TECHNICAL UNIVERSITY, UTTAR PRADESH LUCKNOW

Evaluation Scheme & Syllabus

For

B.Tech in Mathematics and Computing

Second Year

(Effective from the Session: 2026-27)

NOIDA INSTITUTE OF ENGINEERING & TECHNOLOGY, GREATER NOIDA
(An Autonomous Institute)

Mathematics and Computing
Evaluation Scheme
SEMESTER III

Sl. No.	Subject Codes	Subject Name	Subject Type	Periods			Evaluation Schemes			End Semester			Total	Credit
				L	T	P	CT	TA	TOTAL	PS	TE	PE		
		WEEKS COMPULSORY INDUCTION PROGRAM												
1	CCSCC0301	Employability Skill Development-I	Mandatory	2	0	0	60	40	100				100	2
2	CAS0301	Statistics and Probability	Mandatory	3	1	0	30	20	50		100		150	4
3	CCSE0303A	Operating Systems	Mandatory	2	0	0	30	20	50		50		100	2
4	CCSE0301	Data Structures and Algorithm-II	Mandatory	3	0	0	30	20	50		100		150	3
5	CAS0303	Numerical Methods and Optimization	Mandatory	2	0	0	30	20	50		50		100	2
6	CCSE0304	Computer Architecture & Parallel Processing	Mandatory	3	0	0	30	20	50		100		150	3
7	CCSE0353A	Operating Systems Lab	Mandatory	0	0	4				50		50	100	2
8	CCSE0351	Data Structures and Algorithm-II Lab	Mandatory	0	0	4				50		50	100	2
9	CCSDS0351	Foundations of Data Science Lab	Mandatory	0	0	2				25		25	50	1
10	CCSE0355	Object Oriented Techniques using Java	Mandatory	0	0	6				50		100	150	3
11	CCSE0359	Internship-I	Mandatory	0	0	2				50			50	1
12	CNC0301/ CNC0302	Artificial Intelligence and Cyber Ethics / Enviromental Science	Mandatory	2	0	0	30	20	50				50	NA
13		MOOCs (For B.Tech. Hons. Degree)	*MOOCs											
		TOTAL		**17**	**1**	**18**	**210**	**140**	**350**	**225**	**400**	**225**	**1200**	**25**

*** List of MOOCs (Infosys Springboard) Based Recommended Courses for Second year B. Tech Students**

S. No.	Subject Code	Course Name	University/ Industry Partner Name	N. of Hours	Credits
1.	CMC0044	Getting Started with Linux Fundamentals 9.1	RedHat	6h	1
2.	CMC0028	Probability and Statistics using Python	Infosys Wingspan (Infosys Springboard)	16h	1
3	CMC0015	Data Structures and Algorithms using Python - Part 1	Infosys Wingspan (Infosys Springboard)	29h 27m	2
4	CMC0027	Object Oriented Programming Using Python	Infosys Wingspan (Infosys Springboard)	46h 13m	3.5

PLEASE NOTE: -

- **A 3–4-week Internship shall be conducted during summer break after semester-II and will be assessed during semester-III.**
- **Compulsory Audit (CA) Courses (Non-Credit - CNC0301/CNC0302)**
  - All Compulsory Audit Courses (a qualifying exam) do not require any credit.
  - The total and obtained marks are not added to the grand total.

Abbreviation Used:

L: Lecture, T: Tutorial, P: Practical, CT: Class Test, TA: Teacher Assessment, PS: Practical Sessional, TE: Theory End Semester Exam.
CE: Core Elective, OE: Open Elective, DE: Departmental Elective, PE: Practical End Semester Exam, CA: Compulsory Audit,
MOOCs: Massive Open Online Courses.

NOIDA INSTITUTE OF ENGINEERING & TECHNOLOGY, GREATER NOIDA

(An Autonomous Institute)

Mathematics and Computing

Evaluation Scheme

SEMESTER IV

Sl. No.	Subject Codes	Subject Name	Subject Type	Periods			Evaluation Schemes				End Semester		Total	Credit
				L	T	P	CT	TA	TOTAL	PS	TE	PE		
1	CASCC0401	Employability Skill Development – II	Mandatory	2	0	0	60	40	100				100	2
2	CCSE0401	Database Management Systems	Mandatory	2	0	0	30	20	50		50		100	2
3	CMCT0401	Quantum Information Theory	Mandatory	3	0	0	30	20	50		100		150	3
4	CASMCT0401	Applied Linear Algebra and Quantum Mechanics	Mandatory	3	0	0	30	20	50		100		150	3
5		Departmental Elective-I	Departmental Elective	3	0	0	30	20	50		100		150	3
6	CASL0401	Technical Communication	Mandatory	2	0	0	30	20	50		50		100	2
7	CCSE0451	Database Management Systems Lab	Mandatory	0	0	4				50		50	100	2
8	CMCT0451	Quantum Information Theory Lab	Mandatory	0	0	2				25		25	50	1
9	CCSE0455	Web Technologies	Mandatory	0	0	6				50		100	150	3
10	CCSE00459	Mini Project	Mandatory	0	0	2				50			50	1
11	CCSCC0452	Problem Solving Approaches	Mandatory	0	0	2				50			50	1
12	CNC0402/ CNC0401	Environmental Science / Artificial Intelligence and Cyber Ethics	Mandatory	2	0	0	30	20	50				50	NA
13		MOOCs (For B.Tech. Hons. Degree)	*MOOCs											
		TOTAL		**17**	**0**	**16**	**210**	**140**	**350**	**225**	**400**	**175**	**1150**	**23**

*** List of MOOCs (Infosys Springboard) Based Recommended Courses for Second year B. Tech Students**

S. No.	Subject Code	Course Name	University/ Industry Partner Name	N. of Hours	Credits
1	CMC0075	Comprehensive Training on Unix and Linux OS Fundamentals	Infosys Wingspan (Infosys Springboard)	29h53m	2
2	CMC0076	Building Machine Learning Systems with TensorFlow	Infosys Wingspan (Infosys Springboard)	27h 18m	2
3	CMC0077	RDBMS for Data Science	NASSCOM	30 hours	2
4	CMC0065	Data Science for Beginners	NASSCOM	30h	2.5

PLEASE NOTE: -

- **Compulsory Audit (CA) Courses (Non-Credit - CNC0401/CNC0402)**
  - All Compulsory Audit Courses (a qualifying exam) do not require any credit.
  - The total and obtained marks are not added to the grand total.

Abbreviation Used:

L: Lecture, T: Tutorial, P: Practical, CT: Class Test, TA: Teacher Assessment, PS: Practical Sessional, TE: Theory End Semester Exam.
CE: Core Elective, OE: Open Elective, DE: Departmental Elective, PE: Practical End Semester Exam, CA: Compulsory Audit,
MOOCs: Massive Open Online Courses.

NOIDA INSTITUTE OF ENGINEERING & TECHNOLOGY, GREATER NOIDA

(An Autonomous Institute)

List of Department Elective (if any):-

S.No.	Subject Code	Subject Name	Bucket Name	Semester
1	CCSDS0411	Fundamentals of Cloud Computing	Data Analytics	IV
2	CCSE0411	Python web development with Django	Full Stack	IV

NOIDA INSTITUTE OF ENGINEERING & TECHNOLOGY, GREATER NOIDA
(An Autonomous Institute)

AICTE Guidelines in Model Curriculum:

A student will be eligible to get Under Graduate degree with Honours only, if he/she completes the additional MOOCs courses such as Infosys Springboard/NASSCOM/CISCO (as applicable) certifications, or any other online courses recommended by the Institute (Equivalent to 20 credits). During Complete B.Tech. Program Guidelines for credit calculations are as follows.

1. For 6 to 12 Hours =0.5 Credit
2. For 13 to18 =1 Credit
3. For 19 to 24 =1.5 Credit
4. For 25 to 30 =2 Credit
5. For 31 to 35 =2.5 Credit
6. For 36 to 41 =3 Credit
7. For 42 to 47 =3.5 Credit
8. For 48 and above =4 Credit

For registration to MOOCs Courses, the students shall follow Infosys Springboard/NASSCOM/CISCO (as applicable) registration details as per the assigned login and password by the Institute these courses may be cleared during the B. Tech degree program (as per the list provided). After successful completion of these MOOCs courses, the students shall provide their successful completion status/certificates to the Controller of Examination (COE) of the Institute through their coordinators/Mentors only.

The students shall be awarded Honours Degree as per following criterion.

i. If he / she secures 7.50 as above CGPA.
ii. Passed each subject of that degree program in the single attempt without any grace.
iii. Successful completion of MOOCs based 20 credits

Course Code: CCSCC0301	Course Name: Employability Skill Development - I	L	T	P	C
Course Offered in: B. Tech. 2nd Year (3rd Semester) Note: Section C - offered in B.Tech. III/IV SEM		0	0	4	2

Pre-requisite: Students should have successfully completed communication skills courses in the first year.

Course Objectives:

Course Objectives:

Section A: Develop understanding of computer systems, mathematical foundations, and programming concepts while applying structured problem-solving techniques to design logical solutions effectively.

Section B: Enhance numerical ability and logical reasoning skills through speed mathematics, number systems, and analytical problem-solving techniques for competitive aptitude development.

Section C: To develop the ability to communicate effectively and empathetically in workplace situations while strengthening active listening, adaptability, and cross-cultural awareness for navigating complex professional environments.

Course Outcome: After completion of the course, the student will be able to		**Bloom's Knowledge Level (KL)**
CO1	Apply fundamental concepts of computer systems, mathematical foundations, and programming constructs to analyze and solve basic computational problems.	**K3**
CO2	Design and develop structured solutions for problem statements by applying SDLC principles, logical thinking, and flowchart/pseudocode techniques.	**K6**
CO3	Apply number theory and logical reasoning concepts to solve quantitative and analytical problems efficiently.	**K3**
CO4	Apply spoken English skills to participate effectively in standard workplace routines, including daily stand-ups, reporting, and collaborative problem-solving.	**K3**
CO5	Evaluate and adapt communication styles to suit dynamic workplace communication situations and deliver context-sensitive, and logically structured verbal responses under time constraints.	**K6**

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	-	2	3	1	-	3	2	2	2	1	1	2
CO2	3	3	3	3	3	2	1	3	2	2	2	2	2	2
CO3	3	3	2	1	-	-	-	1	2	2	2	3	2	2
CO4	-	-	-	-	1	2	-	3	2	2	-	-	-	-
CO5	-	2	1	-	1	2	-	3	2	2	1	-	-	-

Course Contents / Syllabus

Section A

Module-1	**Foundations of Computer Systems and Mathematical Concepts**	**08 hours**

Computer System Fundamentals: Introduction to Assembler, Compiler, Interpreter, Role of Loader and Linker in program execution.

Mathematical Foundations for Computing: Sets, Relations, and Functions: definitions and applications, Principle of Mathematical Induction and its use in proofs.

Basic Concepts of Programming Language: Memory model of C/C++ Program, Activation Record of a function, L-Value/ R-Value concept. Operators, Precedence, Order of Evaluation and Associativity. Pointers and Dynamics Memory Concept, Functions (argument passing, concept of scope), Understanding Recursion, Bitwise Operations.

Module-2	**Design and develop structured solutions for problem statements**	**08 hours**

Introduction to Software Development Life Cycle, Step-by-step solution to simple problems, Developing logic/flowchart/pseudocode. Stepwise refinement and Procedural Abstraction.

Case Study*: Understand the problem statement, identify a feasible solution, and develop the logic along with a flowchart for the proposed solution.

Section B

Module-3	**Speed Math and Number System**	**16 hours**

Classification of number, Divisibility Rule, Factorization, HCF & LCM, It's Application, Unit digit (Cyclicity), Last two digit, Remainder theorem, Factorial and Number of zeroes, Highest power.

Analytical and Logical Reasoning: Direction and Sense, Blood Relation, Number Series and Letter Series, Coding Decoding

Section C

Module-4	**Workplace Interaction Essentials**	**12 hours**

- The Professional Shift: Navigating the transition from academic to professional tone. Using appropriate register when addressing peers vs. supervisors.
- Active Listening & The Feedback Loop: Moving beyond passive hearing. Practicing the SBI (Situation-Behaviour-Impact) model for peer feedback and clarifying multi-step technical instructions.
- Structured Micro-Speaking: The art of the concise update. Delivering 1-2 minute "Stand-up" reports (What I did, What I will do, Blockers) and explaining basic technical processes to laymen.
- Guided Interpersonal Simulations: Role-playing everyday scenarios: politely interrupting a meeting, seeking assistance on a delayed task, and confirming understanding of complex briefs using paraphrasing techniques.

Digital & Hybrid Nuances: Verbal signalling in virtual meetings, managing "overtalk," and translating concise written chat messages into spoken professional updates.

Module-5	**Workplace Communication in Action**	**12 hours**

- Dynamic Problem-Based Communication: Simulating the "War Room." Explaining sudden technical failures, proposing immediate solutions, and negotiating revised deadlines with stakeholders.
- Listening Under Pressure & Global Adaptability: Interpreting fast, accented, or incomplete instructions (incorporating CEFR-level listening benchmarks). Decoding tone, urgency, and indirect cultural communication styles.
- Collaborative Conflict Resolution: Managing disagreements in team settings using "I-statements" and the "Yes, and…" technique to build consensus without escalating tension.
- The Persuasive Professional: Delivering evidence-based arguments using the PREP (Point, Reason, Example, Point) framework to justify a technical choice or resource allocation.
- Real-Time "Crisis" Simulation (Capstone): A comprehensive role-play where student groups are given a sudden project change or client issue. They must conduct an emergency meeting, delegate tasks verbally, and present a united response to the "client."

	Total Lecture Hours	**56 hours**

*A list of problem statements will be provided by the mentor; you may choose any one. Note: Students may choose an alternative problem statement with prior approval from the mentor.

Reference Books:

	Authors Name
1.A Project-Based Introduction to Programming	Access Point Publishing
2.Programming: Principles and Practice Using C++	Bjarne Stroustrup
3.Effective Modern C++	Scott Meyers
4.Quantitative Aptitude	R S Aggarwal
5.Verbal & Non-Verbal Reasoning	R S Aggarwal
6.Crucial Conversations Tools	Kerry Patterson et al.
7.HBR Guide to Persuasive Presentations	Nancy Duarte
Self-Study Content Links:	**Duration (in Hrs)**
MOOCs	
https://swayam-plus.swayam2.ac.in/courses/course-details?id=F_WADHW_042	**105 hours**
Video Lectures	
Employability Skills Training Session - YouTube	**1 hour**
Web References	
https://dde.manuu.edu.in/sites/default/files/DDE/DDE-Assignments/Diploma-Sem1-2024/Employability-Skills-1.pdf	

Course Code: CAS0301	**Course Name:** Statistics and Probability	**L**	**T**	**P**	**C**
Course Offered in: B. Tech. Second Year: Semester-III/IV AI/AIML/AI (TWIN)/AIML(TWIN)/CYS/DS/CS/CSE/CSE(TWIN)/CSE R/IT/ IT- Twin/ M. Tech. (Int.)/MCT		**3**	**1**	**0**	**4**

Pre-requisite: B. Tech. First Year Syllabus

Course Objectives:

1. The objective of this course is to familiarize the students with concepts of Probability and statistical techniques.
2. It aims to equip the students with adequate Knowledge of statistics that will enable them in formulating Problems and solving problems analytically

Course Outcome: After completion of the course, the student will be able to		Bloom's Knowledge Level (KL)
CO1	Apply the concept of moments, skewness and kurtosis in relevant field.	K3
CO2	Apply the concept of correlation, regression and curve fitting with real world problems.	K3
CO3	Apply the concept of probability and random variable.	K3
CO4	Apply the concept of Mathematical Expectations and Probability Distribution in real life problems.	K3
CO5	Apply the concept of hypothesis testing and statistical quality control to create control charts.	K3

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	2	2	3	1	1	-	1	2	2	2	3	1	1
CO2	3	2	2	3	1	1	-	1	2	2	2	3	2	2
CO3	3	2	1	2	-	-	-	-	1	2	2	2	1	1
CO4	3	2	2	3	1	1	-	1	2	2	2	3	2	2
CO5	3	2	2	3	1	1	-	1	2	2	2	3	2	2

Course Contents / Syllabus

Module 1	Statistical Techniques-I	**9 hours**

Introduction: Measures of central tendency: Mean, Median, Mode, Standard deviation, Quartile deviation, Moments, Skewness, Kurtosis.

Module 2	Statistical Techniques-II	**9 hours**

Curve Fitting: Straight line, Second-degree parabola, Exponential curve by method of least squares, Correlation and Rank correlation, Linear regression, nonlinear regression and multiple linear regression.

Module 3	Probability and Random Variable	**9 hours**

Random Variable: Definition of a Random Variable, Discrete Random Variable, Continuous Random Variable, Probability mass function, Probability Density Function, Distribution functions.
Bivariate Random Variables: Joint probability mass function, joint density and distribution Function, Properties of Joint Distribution function, Marginal Probability mass function, Marginal density Functions, Conditional Distribution for Bivariate Random Variables, Statistical Independence, Central Limit Theorem (Proof not expected)

Module 4	Expectation and Probability Distributions	**9 hours**

Expectations of single Random Variable, Mean, Variance, Moment Generating Function, Binomial, Poisson, Normal, Exponential distribution.		
Module 5	Hypothesis Testing and Control Charts	**9 hours**
Testing a Hypothesis, Null hypothesis, Alternative hypothesis, Level of significance, Confidence limits, Test of significance of means by Z-test and t-test, Chi-square test, F-test, One way ANOVA. Statistical Quality Control (SQC), Control Charts, Control Charts for variables (Mean and Range Charts), Control Charts (p, np and C charts)		
	Total Lecture Hours	**45hours**

Textbook:

1. Textbook of Engineering Mathematics- IV Bali, N.P.
2. Advanced engineering mathematics Jain, R.K.
3. Higher engineering mathematics Grewal, B.S.
4. Statistical methods Gupta, S.P.
5. Advanced engineering mathematics ZILL, DENNIS G.

Reference Books:

1. Introduction to Probability Models Ross, Sheldon M
2. Probability, Random Variables and Stochastic Processes Papoulis, Athanasios
3. Advanced engineering mathematics Kreyszig, E

Self-Study Content Links:		Duration (in Hrs)
MOOCs	https://www.futureskillsprime.in/course/probability-and-statistics/	30 hrs
Video Lectures	https://onlinecourses.nptel.ac.in/e-learning/preview/noc21_ma74 https://onlinecourses.nptel.ac.in/e-learning/preview/noc23_ma77 https://onlinecourses.nptel.ac.in/e-learning/preview/noc23_mg10	13 hrs
Web Reference	https://www.geeksforgeeks.org/maths/probability-and-statistics/	3 hrs
	https://www.geeksforgeeks.org/machine-learning/statistics-for-machine-learning/	2 hrs

Course Code: CCSE0303A	Course Name: Operating Systems	L	T	P	C
Course Offered in: B. Tech. 2nd Year (3rd Semester) CSE/CSE-R/IT/CS/AI/AIML/ IOT/DS/CYS		2	0	0	2

Pre-requisite: Basic knowledge of computer fundamentals, C Programming, Data structures and Computer organization.

Course Objectives:

The objective of the course is to provide a foundational understanding of operating system concepts, including system architecture, process and thread management, concurrency, deadlock, resource management, memory and file systems, Linux shell scripting, and an introduction to virtualization and distributed systems.

Course Outcome: After completion of the course, the student will be able to		Bloom's Knowledge Level (KL)
CO1	Understand operating system architecture and types, and use the Linux CLI for basic Operations.	K2
CO2	Implement the CPU scheduling algorithms including uses of multithreading models.	K3
CO3	Implement concurrency control, process synchronization techniques, and deadlock handling techniques.	K3
CO4	Implement memory management strategies to optimize system performance and analyze file systems and configure virtual machines in modern operating systems.	K3

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	2	2	1	2	-	-	-	-	1	1	2	1	2
CO2	3	3	3	2	2	-	-	-	-	1	1	2	2	1
CO3	3	3	3	2	2	-	-	-	-	1	1	2	3	3
CO4	3	3	3	2	2	-	-	-	-	1	2	2	2	2

Course Contents / Syllabus

Module 1	Fundamentals & Shell scripting	08 hours

Fundamentals of Operating Systems: Overview of Operating Systems, Generations of OS, Operating system architecture, Interrupt handling, System call and kernel, Types of Operating System: Batch OS, Multi-programming OS, Multitasking OS, Multiprocessor OS, Real time OS.

Shell Scripting in Linux: Introduction to Linux Operating System & Architecture, Basic Command Line Interface (CLI) Operations in Linux, Shell Scripting Basics: Variables, Control Structures, Functions.

Case study: Automating system administration tasks using shell scripts in Linux.

Module 2	Process & Thread Management	08 hours

Process Management: Process, Transition Diagram, Process Control Block (PCB), Types of Schedulers: Long Term, Mid Term, Short Term Scheduler,

CPU Scheduling: Pre-emptive and Non-Pre-emptive Algorithm (FCFS, SJF, SRTF, Non-Pre-emptive Priority, Pre-emptive Priority, Round Robin, Multilevel Queue Scheduling and Multilevel Feedback Queue Scheduling)

Thread: Processes Vs Threads, Thread states, Benefits of threads, Types of threads, Multithreading Models, Concept of Hyper-Threading

Case study: Analyse and implement CPU Scheduling in Real-Time Embedded Systems and RTOS.

Module 3	Concurrency and Deadlock Management	08 hours

Concurrency: Introduction of Concurrency, Types of Process, Race Condition, Critical Section, Inter Process Communication, Producer consumer problem.

Process Synchronization: Lock variable, Peterson's Solution, Strict alternation, Lamport Bakery Solution, Test and set lock, Semaphore- counting, binary and monitor,

Classical Problem of Synchronization: Bound Buffer, Dinning Philosopher, Reader writer, Sleeping barber.

Deadlock: Deadlock, Deadlock characterization, Deadlock Prevention, Deadlock Avoidance: Bankers Algorithms, Deadlock Detection, Recovery from Deadlock.

Case study: Deadlock avoidance in database transaction management systems like Oracle or MySQL.

Module 4	Memory Management	08 hours

Memory Management: Memory Management function, Loading and linking Address Binding, Memory management techniques, Contiguous technique- Fixed Partitions, variable partitions, Memory Allocation: Allocation Strategies (First Fit, Best Fit, and Worst Fit), Non-contiguous, Paging, Segmentation, Segmented paging,

Virtual Memory: Virtual Memory Concepts, Demand Paging, Performance of Demand Paging, Page Replacement Algorithms: FIFO, LRU, Optimal and LFU, Belady's Anomaly, Thrashing.

File and Disk Management: File Access Mechanism, File Allocation Method, Disk Scheduling Algorithms

Modern Operating System: Virtual machines – hypervisor, Introduction to GPU. Introduction of CUDA Programming.

Case study: Virtual memory management in modern OS like Windows 10 and how paging impacts performance.

Total Lecture Hours	**32 hours**

Textbooks:

	Authors Name
1." Operating System Concepts Essentials", Willey Publication,10th Edition,2018.	Abraham Silberschatz, Peter Baer Galvin and Greg Gagne
2."A practical guide to Linux: Commands, Editors and Shell Programming", CreateSpace Independent Publishing Platform, 4th Edition,2017.	Marks G. Sobell
3."LINUX for beginners", 1stEdtion,2014.	Jason Cannon

Reference Books:

	Authors Name
1."Operating Systems: Internals and Design Principles", Pearson Education , 9th Edition, 2019.	William Stallings
2."Operating System: A Design-oriented Approach" , McGraw Hill Education,2017.	Charles Patrick Crowley
3."Learning Linux Shell Scripting", Packt Publishing ,2nd Edition 2018.	Ganesh Naik

Self-Study Content Links:		**Duration (in Hrs)**
Video Lectures	https://www.youtube.com/watch?v=dTl9QkH4j8o&list=PL6CdojO56mZ3SeRfpzMBMObSnTziA0gfE https://www.youtube.com/watch?v=Q35sMYHaIvQ https://www.youtube.com/watch?v=rnGVincwk30 https://www.youtube.com/watch?v=akU1Ji8Vzdk&list=PLbMVogVj5nJRa3VKt_eyZdJDitCz1cvQ	21Hrs
	https://www.youtube.com/watch?v=pPAKs7tT8sw%20 https://www.youtube.com/watch?v=hBPYP0ZEvS8%20 https://www.youtube.com/watch?v=UeVQxzlgd10%20 https://www.youtube.com/watch?v=9cuwgiHQyoI%20 https://www.youtube.com/watch?v=-NONm-Jq34Y&t=25s	3Hrs
	https://www.youtube.com/watch?v=5uyGn147rr0%20 https://www.youtube.com/watch?v=utYC2tKDdoQ%20 https://www.youtube.com/watch?v=Cuo8GTklgoE%20 https://www.youtube.com/watch?v=YHQxp-XduS0%20 https://www.youtube.com/watch?v=yB8sm-pwvZo%20 https://www.youtube.com/watch?v=Y14b7_T3AEw	6Hrs
	https://www.youtube.com/watch?v=bK-VhQA512c%20 https://www.youtube.com/watch?v=o2_iCzS9-ZQ%20 https://www.youtube.com/watch?v=SnKgEuUfV4k%20 https://www.youtube.com/watch?v=86FAWCzIe_4	16Hrs

Course Code: CCSE0301	Course Name: Data Structures & Algorithms-II	L	T	P	C
Course Offered in: B. Tech. 2nd Year (3rd Semester) CSE/CS/IT/CSE(AIML)/CSE(DS)/CSE(AI)		3	0	0	3

Pre-requisite: C, Python

Course Objectives:

The objective of the course is to learn the basic concepts of algorithm analysis, along with the implementation of non-linear data structures.

Course Outcome: After completion of the course, the student will be able to		Bloom's Knowledge Level (KL)
CO1	Apply tree structures to solve specific problems using various tree operations and algorithms.	K3
CO2	Analyse the graph data structure and evaluate the efficiency of its operations for problem solving.	K4
CO3	Evaluate dynamic programming solutions for efficient problem-solving across diverse contexts.	K4
CO4	Apply efficient backtracking and branch –and –bound techniques across diverse problem-solving scenarios.	K3
CO5	Understand principles of advanced data structures, including their implementation and applications.	K2

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	2	1	1	2	1	1	2	1	-	2	1	2
CO2	3	3	2	1	1	2	1	1	2	1	-	2	1	2
CO3	3	3	2	2	2	2	1	1	2	2	-	2	1	2
CO4	3	3	3	2	2	2	1	1	2	1	-	2	1	2
CO5	3	3	3	2	2	2	1	1	2	2	-	2	1	2

Course Contents / Syllabus

Unit-1	Design and Analysis of Algorithms: Tree	09 hours

Trees: Terminology used with Trees, Binary Tree, Memory representation of Tree, Traversal Algorithms: In-order, pre-order, and post-order. Constructing Binary Tree from given Tree Traversal, Operation of Insertion, Deletion, Searching & Modification of data in Binary Search tree, Binary Heaps, Threaded Binary trees, Traversing Threaded Binary trees, AVL Tree.
Application of Trees: Priority Queue, Heap Sort.

Unit-2	Design and Analysis of Algorithms: Graphs	09 hours

Graphs: Terminology used with Graph, Data Structure for Graph Representations: Adjacency matrices, Adjacency List.
Graph Traversal: Depth First Search and Breadth First Search. Connected Component, Spanning Trees.
Algorithms on Graphs: Minimum Cost Spanning **Trees:** Prim' s and Kruskal's algorithm, Transitive Closure and Shortest Path algorithms: Dijkstra Algorithm, Bellman Ford Algorithm, Floyd Warshall's Algorithm.

Unit-3	Dynamic Programming	09 hours

Dynamic Programming: Dynamic Programming concepts 0/1 Knapsack, Longest Common Sub Sequence, Matrix Chain Multiplication, Resource Allocation Problem.

Unit-4	Backtracking, Branch and Bound	09 hours

Backtracking: Backtracking, Branch, and Bound with Examples Such as Travelling Salesman Problem, Graph Coloring, n-Queen Problem, Hamiltonian Cycles, and Sum of Subsets.

Unit-5	Advanced- Data Structures	09 hours

Red-Black Trees, B – Trees, B+ Trees, Binomial Heaps, Fibonacci Heaps.

	Total Lecture Hours	45 hours

Textbooks:

	Authors Name
1."Data Structures and Algorithms in Python: An Indian Adaptation", 1st Edition, 2021	Michael T. Goodrich, Roberto Tamassia
2.“Data Structures” Schaum’s Outline Series, Tata McGraw-hill Education (India) Pvt. Ltd, 2nd Edition, 2017.	Lipschutz
3.“Introduction to Algorithms”, Printice Hall of India, 4th Edition, 2022	Thomas H. Coreman, Charles E. Leiserson and Ronald L. Rivest.

Reference Books:

	Authors Name
1. “Data Structure Using C”, Oxford University Press, 2nd Edition, 2014.	Reema Thareja
2. “Data Structure Using C”, Pearson Education India, 2nd Edition,2011.	AK Sharma
3.“C and Data structure”, Wiley Dreamtech Publication, 1st Edition, 2004.	P. S. Deshpandey

Self-Study Content Links:		**Duration (in Hrs)**
MOOCs	Computational Problem Solving	29Hrs 02Min
Video Lectures	https://youtu.be/u5AXxR4GnRY https://www.youtube.com/watch?v=LQx9E2--p5c&pp=ygUMYXJyYXlzIG5wdGVs https://www.youtube.com/watch?v=K7VIKlUdo20&pp=ygUPbGluayBsaXN0IG5wdGVs https://www.youtube.com/watch?v=g1USSZVWDsY&list=PLB3CD0BBB95C1BF09&index=2&pp=iAQB https://www.youtube.com/watch?v=THMyk2_p530&pp=ygUccXVldWUgZGF0YSBzdHJ1Y3R1cmUgICBucHRlbA%3D%3D https://www.youtube.com/watch?v=_VV9v41FIq0&pp=ygUZZGl2aWRlIGFuZCBjb25xdWVyICBucHRlbA%3D%3D https://www.youtube.com/watch?v=ARvQcqJ_-NY&list=PLfFeAJ-vQopt_S5XlayyvDFL_mi2pGJE3	00Hrs 02Min 00Hrs 14Min 00Hrs 42Min 01Hrs 04Min 01Hrs 01Min
Web References	https://www.udemy.com/course/data-structures-and-algorithms-java/?couponCode=PMNVD2025 https://www.coursera.org/courses?query=data%20structures%20and%20algorithms https://www.codechef.com/roadmap/data-structures-and-algorithms	

Course Code: CAS0303	**Course Name:** Numerical Methods and Optimization	L	T	P	C
Course Offered in: Department of Mathematics and Computing, B.Tech. III Sem		2	0	0	2

Pre-requisite: Calculus, Linear Algebra, and Basic Programming Knowledge

Course Objectives:

This course aims to introduce the fundamental principles of optimization and numerical methods, enabling students to model and solve real-world engineering problems. It emphasizes the application of numerical techniques for approximate solutions, fosters understanding of stability, convergence, and error estimation, and develops computational thinking using relevant software tools.

Course Offered in: Department of Mathematics and Computing, B.Tech. III Sem

Course Outcome: After completion of the course, the student will be able to		**Bloom's Knowledge Level (KL)**
CO1	Solve basic optimization problems using classical techniques.	K3
CO2	Apply numerical methods for solving algebraic and transcendental equations.	K3
CO3	Demonstrate the ability to solve systems of linear equations and perform interpolation.	K3
CO4	Apply techniques of numerical integration and differentiation.	K3
CO5	Analyze numerical algorithms using appropriate software tools.	K4

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	
CO1	3	3	3	2	3	-	-	-	1	-	2	3	3	
CO2	3	3	3	2	3	-	-	-	1	-	2	3	3	
CO3	3	2	3	3	3	-	-	-	1	-	2	3	3	
CO4	3	3	2	3	3	-	-	-	1	-	2	3	3	
CO5	3	3	3	3	3	-	-	-	1	-	2	3	3	

Course Contents / Syllabus

Module 1	**Optimization Techniques: Introduction and Linear Programming**	**9 hours**
Introduction to optimization and types of optimization problems; Formulation of Linear Programming Problems (LPP); Graphical Method and Simplex Method; Duality and Sensitivity Analysis; Applications in engineering and decision science		
Module 2	**Non-Linear and Unconstrained Optimization**	**7 hours**
Unconstrained Optimization Techniques: Gradient Descent, Newton's Method; Constrained Optimization using Lagrange Multipliers; KKT Conditions; Introduction to Convex Optimization; Numerical solutions using software tools		
Module 3	**Numerical Solution of Equations and Interpolation**	**7 hours**
Solving Algebraic and Transcendental Equations: Bisection, Regula-Falsi, Newton-Raphson; Rate of Convergence and Error Estimation; System of Linear Equations: Gauss Elimination, Gauss-Jordan, LU Decomposition; Iterative Methods: Gauss-Seidel, Jacobi, Newton's Interpolation, Lagrange Interpolation; Spline Interpolation; Least Squares Method for Curve Fitting		
Module 4	**Numerical Integration and Differentiation**	**7 hours**
Finite Differences, Forward and Backward Differences; Numerical Differentiation: Finite Difference Approximations; Numerical Integration: Trapezoidal Rule, Simpson's Rule; Gaussian Quadrature; Numerical Solution of Ordinary Differential Equations: Euler's Method, Runge-Kutta Methods		
Total Lecture Hours		**30 hours**

Textbook:

S.No	Book Title	Author
1	Optimization: Theory and Applications, New Age International. 2nd edition,1984	S. S. Rao
2	Elementary Numerical Analysis, McGraw Hill. 3rd edition, 1980	S. D. Conte and Carl de Boor

Reference Books:

S.No	Book Title	Author
1	Operations Research, Sultan Chand & Sons. 20th edition, 2022	Kanti Swarup, P. K. Gupta, Man Mohan,
2	Numerical Methods, McGraw Hill. 5th edition, 2019	E. Balagurusamy,
3	Applied Numerical Methods with Python for Engineers and Scientists, McGraw Hill. 1st edition 2021	Steven C. Chapra,

NPTEL/ Youtube/ Faculty Video Link:

Module 1:	NPTEL – Optimization Techniques by Prof. S. S. Rao (https://nptel.ac.in)
Module 2:	YouTube – Convex Optimization Basics (https://www.youtube.com)
Module 3:	NPTEL – Numerical Methods for Engineers (https://nptel.ac.in), YouTube – Interpolation Techniques (https://www.youtube.com)
Module 4:	NPTEL – Numerical Integration and ODEs (https://nptel.ac.in)

LAB Course Code: CAS0353	**LAB Course Name:** Numerical Methods and Optimization Lab	L	T	P	C
Course Offered in: B.Tech Mathematics and Computing (Quantum Computing Specialization)		0	0	2	1

Pre-requisite: Basic Python programming knowledge, numerical methods, and calculus

Course Objectives:

This course objective is to provide hands-on experience in implementing optimization and numerical techniques using Python, fostering skills in mathematical modeling, algorithmic thinking, and the use of Python libraries to solve real-world problems programmatically.

Course Outcome: After completion of the course, the student will be able to		Bloom's Knowledge Level (KL)
CO1	Implement classical optimization techniques using Python.	K2
CO2	Write programs to solve linear and non-linear equations numerically.	K4
CO3	Develop Python solutions for numerical integration and interpolation.	K2
CO4	Solve systems of linear equations using matrix-based numerical methods.	K4
CO4	Use libraries like NumPy, SciPy, and Matplotlib for mathematical computing.	K4

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO1	3	3	3	2	3	-	-	-	2	-	2	3	3
CO2	3	3	3	2	3	-	-	-	2	-	2	3	3
CO3	3	3	3	3	3	-	-	-	2	-	2	3	3
CO4	3	3	3	3	3	-	-	-	2	-	2	3	3
CO5	3	3	3	3	3	-	-	-	2	-	2	3	3

List Of Practical's (Indicative & Not Limited To)

Objective	**CO Mapping**
1. Solve linear programming problems using the Simplex method with the help of SciPy library.	CO1, CO5
2. Perform unconstrained optimization using gradient descent method and visualize convergence.	CO1, CO5
3. Perform root finding using Bisection and Secant methods and compare results graphically.	CO2, CO5
4. Implement the Newton-Raphson method in Python to find roots of a non-linear equation.	CO2
5. Write a Python script to interpolate using Lagrange's method and plot the result.	CO3, CO5
6. Develop a program to fit a polynomial using Least Squares method and visualize the curve.	CO3, CO5
7. Implement Trapezoidal and Simpson's 1/3 Rule for numerical integration in Python.	CO3
8. Write a Python function to solve ODEs using Euler and Runge-Kutta (RK4) methods.	CO3, CO5
9. Write a Python program to solve a system of linear equations using Gauss Elimination method.	CO4
10. Use NumPy to compute eigenvalues and eigenvectors of a matrix and interpret the results.	CO4, CO5
	Total Hours: 48 hrs.

Course Code: CCSE0304	**Course Name:** Computer Architecture & Parallel Processing	**L**	**T**	**P**	**C**
Course Offered in: B. Tech CSE/CSE-R/IT/CS/AI/AIML/ IOT/DS/CYS		**3**	**0**	**0**	**3**

Pre-requisite: Basic knowledge of computer systems, Logic gates and their operations.

Course Objectives:

This course aims to provide a comprehensive understanding of computer organization and architecture, covering processor design, memory systems, and control units. It explores advanced topics such as cache coherence, parallel architectures, and scalable shared memory systems. Students will gain insight into system performance through instruction execution, arithmetic operations, synchronization, and interconnect strategies.

Course Outcome: After completion of the course, the student will be able to		Bloom's Knowledge Level (KL)
CO1	Understand the basic structure and operation of a digital computer system.	K2
CO2	Analyze the design of arithmetic & logic unit and understand the fixed point and floating-point arithmetic operations.	K4
CO3	Implement control unit techniques and the concept of Pipelining.	K3
CO4	Analyze parallel architectures and coherence protocols by exploring memory hierarchy, cache coherence mechanisms, and multiprocessor design techniques to ensure correctness and performance in parallel systems.	K4
CO5	Analyze scalable shared memory systems by evaluating directory coherence protocols, memory consistency models, synchronization mechanisms, and interconnect strategies to ensure system	K5

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	2	1	1	2	1	–	1	–	–	1	3	1	1
CO2	3	3	1	2	2	1	–	1	–	1	1	3	2	1
CO3	3	2	2	2	2	1	1	1	1	1	2	3	2	2
CO4	3	3	3	3	2	1	1	1	2	1	2	3	3	2
CO5	3	2	2	1	2	1	1	1	1	1	2	2	3	2

Course Contents / Syllabus

Unit 1	**Introduction**	**08 hours**

Computer Organization and Architecture, Functional units of digital system and their interconnections, buses, bus architecture, types of buses and bus arbitration and its types. Register, bus and memory transfer. Processor organization, general registers organization, stack organization and addressing modes.

Unit 2	**ALU Unit**	**08 hours**

Arithmetic and logic unit: Multiplication: Signed operand multiplication, Booth's algorithm and array multiplier. Division and logic operations. Floating point
arithmetic operation, Arithmetic &logic unit design. IEEE Standard for Floating Point Numbers.

Unit 3	**Control Unit**	**08 hours**

Control Unit: Instruction types, formats, instruction cycles and sub cycles (fetch and execute etc.), micro-operations, execution of a complete instruction. Program Control, Reduced Instruction Set Computer, Complex Instruction Set Computer, Pipelining. Hardwire and microprogrammed control, Concept of horizontal and vertical microprogramming, Flynn's classification.

Unit 4	Introduction to Parallel Architectures	08 hours
Introduction to Parallel Architectures, Parallel Programming models and Architectures, Memory Hierarchy-Cache and Virtual memory, Overview of Cache coherence, Coherence Protocols- Snooping, Directory based protocols, VI protocol, MSI, MESI, Dragon protocol and Correctness of coherence protocols-Types of cache misses, update vs invalidate protocol, Snoop based multiprocessor design, Single and multi-level cache with atomic and bus split transaction bus		
Unit 5	**Parallel Systems**	**04 hours**
Scalable shared memory systems: Directory coherence protocols- Memory based, cache based, correctness, Case study: Origin- Architecture, protocol, correctness; Sequent NUMA Q- Architecture, protocol, correctness, Memory consistency models- Sequential, Relaxed consistency models, Synchronization-LL-SC, point to point, barrier synchronization, Interconnects- Introduction, Topologies, routing, flow control		
	Total Lecture Hours	**36 hours**

Textbooks:

	Authors Name
1. "Computer System Architecture", 3rd Edition, Pearson Publication, 2007.	M. Mano
2. Computer Architecture and Organization, Tata McGraw Hill, Third Edition, 1998.	John P. Hayes
3. Computer Architecture: A Quantitative Approach. Morgan- Kaufmann publishers.	J.L. Hennessy and D. A. Patterson
Computer Architecture: A Quantitative Approach. Morgan- Kaufmann publishers.	Michael Nielsen & Isaac Chuang

Reference Books:

	Authors Name
Computer Organization, McGraw-Hill, Fifth Edition, Reprint2012	Carl Hamacher, ZvonkoVranesic, SafwatZaky,
Advanced Microprocessors and Peripherals, TM	Ray A K, Bhurchandi K M ,
"Computer Architecture & Parallel Processing" Mcgraw Hill Education	Kai Hwang
Computer Organization, McGraw-Hill, Fifth Edition, Reprint2012	Carl Hamacher, ZvonkoVranesic, SafwatZaky

Self-Learning/NPTEL/ Youtube/ Faculty Video Link Contact Hrs:		**Duration (in Hrs)**
NPTL	1. https://nptel.ac.in/courses/106105163 2. https://onlinecourses.nptel.ac.in/noc23_cs113/preview 3. https://onlinecourses.nptel.ac.in/noc26_cs89/preview	30Hrs 30Hrs 20Hrs
YouTube	1. https://www.youtube.com/watch?v=L9X7XXfHYdU&list=PLxCzCOWd7aiHMonh3G6QNKq53C6oNXGrX 2. https://www.youtube.com/watch?v=WLgXUPOjKEc 3. https://www.youtube.com/watch?v=BPhWlFIU1rc 4. https://www.youtube.com/watch?v=6R7JDkpG1Wk&list=PLrjkTql3jnm8HbdMwBYIMAd3UdstWChFH 5. https://www.youtube.com/watch?v=nxryfWg5Hm4 https://www.youtube.com/watch?v=txAyA_UozmM	40Hrs 24Hrs 30Min
MOOCs Certificate Course	1. NPTEL – Computer Organization and Architecture By IIT Kharagpur NPTEL Computer Organization and Architecture 2. IBM Quantum Learning Introductory quantum computing and Qiskit tutorials IBM Quantum Learning 3. Coursera – Parallel, Concurrent, and Distributed Programming Offered by Rice University Coursera Parallel Programming Specialization	30Hrs 15 Hrs 40Hrs

ourse Code: CCSE0353A	Course Name: Operating Systems Lab	L	T	P	C
Course Offered in: B. Tech. 2nd Year (3rd Semester) CSE/CSE-R/IT/CS/AI/AIML/ IOT/DS/CYS		0	0	4	2

Pre-requisite: Basic knowledge of computer fundamentals, C programming, Data structures and Computer organization.

Course Objectives:

The course aims to provide hands-on experience with Linux and shell programming, while the lab focuses on implementing and analyzing key OS algorithms and simulating modern operating systems.

Course Outcome: After completion of the course, the student will be able to		Bloom's Knowledge Level (KL)
CO1	Execute basic Linux commands and shell scripts to automate file management and system administration tasks.	K3
CO2	Implement and compare various CPU scheduling algorithms, process synchronization solutions using semaphores and deadlock handling algorithms.	K4
CO3	Simulate memory allocation techniques and page replacement algorithms, disk management strategies and explore modern OS features including virtualization and distributed computing.	K5

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	2	1	1	1	3	-	-	2	2	-	3	2	1	2
CO2	3	3	3	2	2	-	-	2	2	-	2	2	2	3
CO3	3	2	3	2	3	-	-	2	2	-	3	2	3	3

List Of Practical's (Indicative & Not Limited To)

Sr. No	Experiments	CO Mapping
1	Installation of Linux Operating system.	CO1
2	**Implementation of Linux Commands** i. Display system information using uname, hostname, and date etc. ii. File operations using cat, touch, cp, mv, rm, and chmod ,umask etc. iii. Create, view, and navigate directories using mkdir, rmdir, cd, pwd, ls etc. Disk Commands df,du,mount,unmount,mkfs,fsck etc.	CO1
3	**Implementation of Linux Commands (continued)** v. Use redirection and piping in commands. vi. File compression and archiving using tar, gzip, zip, unzip etc. vii. Process commands ps,kill, killall,nice, pgrep, top,htop etc. viii. Network commands ifconfig, ping, netstat, host,ip route etc. Administrator Commands Adduser,Passwd, deluser, usermod, groupadd etc	CO1
4	**Shell Scripting Programming** i. Write a shell script to ask your name, program name and enrollment number and print it on the screen. ii. Write a shell script to find the sum, the average and the product of the four integers entered. iii. write shell script to find average of numbers given at command line iv. Write a shell program to exchange the values of two variables. v. Write a shell program to Print Numbers 1 to 10 using while & do while loop. Write a shell program to Print Numbers 1 to 10 using for loop.	CO1
5	**Shell Scripting Programming (continued)** vii. Write a shell script to display the digits which are in odd position in a given 5-digit number. viii. Write a shell program to search for a given number from the list of numbers provided using binary search method. ix. Write a shell program to concatenate two strings and find the length of the resultant string.	CO1

	x. Write a shell script to find the smallest of three numbers. Write a shell program to count number of words, characters, white spaces and special symbols in a given text.	
6	Implement FCFS CPU Scheduling algorithm.	CO2
7	Implement the SJF CPU Scheduling algorithm (For both Pre-emptive and non-pre-emptive).	CO2
8	Implement PRIORITY CPU Scheduling Algorithm (For both Pre-emptive and non-pre-emptive).	CO2
9	Implement Round-Robin CPU Scheduling Algorithm.	CO2
10	Implement Multi-Level Queue CPU Scheduling algorithm.	CO2
11	Implement Multilevel Feedback Queue CPU Scheduling Algorithm.	CO2
12	Execute the RACE Condition of Process Synchronization.	CO2
13	Implement the Producer–consumer problem using semaphores.	CO2
14	Design a code and implement the Dinning Philosopher problem.	CO2
15	Implement Banker's algorithm of Deadlock Avoidance.	CO2
16	Execute an algorithm for Deadlock Detection.	CO2
17	Implement Contiguous memory variable size partition scheme.	CO3
18	Simulate the First-Fit contiguous memory allocation technique.	CO3
19	Simulate the Best-Fit contiguous memory allocation technique.	CO3
20	Simulate the Worst-Fit contiguous memory allocation technique.	CO3
21	Implement the Non-contiguous Memory Allocation by using Paging.	CO3
22	Implement Contiguous memory variable size partition scheme.	CO3
23	Write a Program to simulate the FIFO page replacement algorithm.	CO3
24	Write a Program to simulate the LRU page replacement Algorithm.	CO3
25	Write a Program to simulate the Optimal page replacement Algorithm.	CO3
26	Write a program to simulate FCFS Disk Scheduling Algorithm.	CO3
27	Write a Program to simulate the SSTF Disk Scheduling Algorithm	CO3
28	Write a program to simulate SCAN Disk Scheduling Algorithm.	CO3
29	Write a Program to simulate the C SCAN Disk Scheduling Algorithm.	CO3
30	Write a Program to simulate the LOOK Disk Scheduling Algorithm.	CO3
	Total Lecture Hours	**30 hours**

LAB Course Code: CCSE0351	Course Name: Data Structures & Algorithms-II Lab	L	T	P	C
Course Offered in: B. Tech. 2nd Year (3rd Semester) CSE/CS/IT/CSE(AIML)/CSE(DS)/ CSE(AI)/CYS		0	0	4	2

Pre-requisite: C, Python

Course Objectives:

Learn to implement non-linear data structures.

Course Outcome: After completion of the course, the student will be able to		Bloom's Knowledge Level (KL)
CO1	Implementation of tree data structures for basic operations like insertion, deletion, searching and traversal.	K3
CO2	Implementation of algorithms based on graph data structures for solving real world problems.	K3
CO3	Implementing Dynamic Programming, Backtracking, Branch and Bound algorithms to solve complex data efficiently and effectively.	K3

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	2	1	1	2	1	1	2	1	--	2	1	2
CO2	3	3	2	1	1	2	1	1	2	1	--	2	1	2
CO3	3	3	2	2	2	2	1	1	2	1	--	2	1	2

List Of Practical's

S. No.	Name of Experiments/ Programs	CO Mapping
1	Write a program to implement an in-order traversal of a binary tree and print the nodes.	CO1
2	Write a program to implement a pre-order traversal of a binary tree and print the nodes.	CO1
3	Write a program to implement a post-order traversal of a binary tree and print the nodes.	CO1
4	Write a program to count the number of nodes in a binary tree.	CO1
5	Write a program to find the height of the tree.	CO1
6	Write a program to check if the binary tree is balanced or not.	CO1
7	Write a program to search a number in a Binary Search Tree (BST).	CO1
8	Write a program to insert a node in a Binary Search Tree (BST).	CO1
9	Write a program to delete a node from a Binary Search Tree (BST).	CO1
10	Write a program to implement a max-heap and perform heap sort on an array of integers.	CO1
11	Write a program to implement a priority queue using a max heap.	CO1
12	Write a program to create a graph using an adjacency matrix.	CO2
13	Write a program to create a graph using an adjacency list.	CO2
14	Write a program to perform Depth-First Search (DFS) on a graph.	CO2
15	Write a program to perform Breadth-First Search (BFS) on a graph.	CO2
16	Write a program to check if there is a path between two nodes in a graph using DFS.	CO2
17	Write a program to find all the vertices reachable from a given vertex in a graph.	CO2
18	Write a program to detect a cycle in an undirected graph using DFS.	CO2

19	Write a program to detect a cycle in a directed graph using DFS.	CO2
20	Write a program to find the degree of each vertex in an undirected graph.	CO2
21	Write a program to count the number of connected components in an undirected graph.	CO2
22	Write a program to implement Dijkstra's Algorithm.	CO3
23	Write a program to implement Prim's Algorithm.	CO3
24	Write a program to implement Kruskal's Algorithm.	CO3
25	Write a program to implement Floyd–Warshall's all-pairs shortest path algorithm.	CO3
26	Write a program to implement Bellman–Ford Algorithm.	CO3
27	Write a program to implement Longest Common Subsequence (LCS) using Dynamic Programming.	CO3
28	Write a program to implement 0/1 Knapsack using Dynamic Programming.	CO3
29	Write a program for Matrix Chain Multiplication using Dynamic Programming.	CO3
30	Write a program to implement Sum of Subsets problem using backtracking.	CO3
		Total Hours: 30 hrs.

Course Code: CCSE0355	Course Name: Object Oriented Techniques using Java	L	T	P	C
Course Offered in: B. Tech. 2nd Year (3rd Semester) CSE/CS/IT/CSE(AI)/ CSE(AIML)/CSE(IOT)/CSE(AI)/CSE(DS)/CSE-R/M.Tech(Int)		0	0	6	3

Pre-requisite: Basic Programming Knowledge, Knowledge of any programming language (e.g., C, C++, Python)

Course Objectives:

The objective of this course is to understand the object-oriented methodology, and its techniques to design stand alone and GUI applications using hands-on engaging activities.

Course Outcome: After completion of the course, the student will be able to		Bloom's Knowledge Level (KL)
CO1	Understand the concepts of object-oriented programming and relationships among them needed in modeling.	K2
CO2	Demonstrate the Java programs using OOP principles and also implement the concepts of lambda expressions	K3
CO3	Analyze packages with different protection level resolving namespace collision and implement the error handling concepts for uninterrupted execution of Java program.	K4
CO4	Analyze Concurrency control, I/O Streams and Java Socket Programming Concepts.	K4
CO5	Design and develop the GUI based application, Generics and Collections in Java programming language to solve the real-world problem	K6

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	3	3	2	2	1	-	1	-	2	3	3	2
CO2	3	3	3	3	2	2	1	-	1	1	2	3	2	2
CO3	3	3	3	3	3	2	2	-	2	1	2	3	3	1
CO4	3	3	3	3	3	2	2	1	2	1	2	3	2	2
CO5	3	3	3	3	3	2	2	1	2	1	2	3	3	2

Course Contents / Syllabus

Unit 1	Basics of Java Programming	16 hours

Object Oriented Programming: Introduction and Pillars of OOP with real life example, jvm architecture and its components
Modelling Concepts: Introduction, Class Diagram and Object Diagram, UML concepts: Association, Composition, aggregation, realization, and Generalization.
Control Statements: Decision Making, Looping and Branching, Argument Passing Mechanism: Command Line Argument, Console Input.
Class and Object: Object Reference, Constructor, Abstraction: Abstract Class, Interface and its uses, Defining Methods, Use of "this" and "super" keyword, Garbage Collection and finalize () Method etc.

Unit 2	OOPs features, arrays and lambda expressions	16 hours

Inheritance: Introduction and Types of Inheritance in Java, Access Modifiers, Constructors and super constructor in Inheritance.
Polymorphism: Introduction and Types, Overloading and Overriding.
Lambda expression: Introduction and Working with Lambda Variables.
Arrays: Introduction and its Types. Jagged Array with example

Unit 3	Packages, Exception Handling and String Handling	16 hours

Packages: Introduction and Types, Access Protection in Packages, Import and Execution of Packages.
Exception Handling, Assertions and Localizations: Introduction and Types, Exceptions vs. Errors, Handling of Exception. Finally, Throws and Throw keyword, Multiple Catch Block, Nested Try and Finally Block, Customized Exceptions, Tokenizer. Assertions and Localizations Concepts and its working.
String Handling: Introduction and Types, Operations, Immutable String, Method of String class, String Buffer and String Builder class.

Unit 4	Concurrency in Java and I/O Stream	16 hours

Threads: Introduction and Types, Creating Threads, Thread Life-Cycle, Thread Priorities, Daemon Thread,Runnable Class, Synchronizing Threads etc.

I/O Stream: Introduction and Types, Common I/O Stream Operations, Interaction with I/O Streams Classes. character and byte oriented stream classes with example.

Java Socket Programming: Introduction and types(TCP, UDP), java socket program with server-side and client-side by using connection.

Unit 5	GUI Programming, Generics and Collections	16 hours

GUI Programming: Introduction and Types, Swing, AWT, Components and Containers, Layout Managers and User-Defined Layout and Event Handling.

Generics: Introduction to Generic Classes, types of generic defined in brief, bounded type parameter(Upper and Lower bound), Initializing a Generic Object, Classes, Methods and Interfaces Use enumerated type.

Collections: Introduction, main interfaces of collections(Collection, List Set, Map, Queue), classes of collections(ArrayList, Linked list, HashSet, HashMap and TreeSet) and methods(List, Set Map) Collection using Iterators.

Total Lecture Hours	**80 hours**

Textbooks:

S.No	Book Title	Author
1	” Java: A Beginner’s Guide”, McGraw-Hill Education 2nd edition	Herbert Schildt
2	“Programming with Java A Primer”, TMH, 4th edition.	E Balagurusamy,

Reference Books:

S.No	Book Title	Author
1	“Core Java Volume I – Fundamentals”, Prentice Hall	Cay S. Horstmann
2	” Effective Java”, Addison Wesley	Joshua Bloch
3	Java - The Complete Reference”, McGraw Hill Education 12th edition	Herbert Schildt

NPTEL/ YouTube/ Faculty Video Link:		Duration (in Hrs)
Unit 1	https://www.youtube.com/watch?v=r59xYe3Vyks&list=PLS1QulWo1RIbfTjQvTdj8Y6yyq4R7g-Al	**13 m 09 s**
Unit 2	https://www.youtube.com/watch?v=ZHLdVRXIuC8&list=PLS1QulWo1RIbfTjQvTdj8Y6yyq4R7g-Al&index=18	**15 m 31 s**
Unit 3	https://www.youtube.com/watch?v=hBh_CC5y8-s	**10h 10 m 57 s**
Unit 4	https://www.youtube.com/watch?v=qQVqfvs3p48	**6 h 20 m 29 s**
Unit 5	https://www.youtube.com/watch?v=2qWPpgALJyw	**5 h 17 m 46 s**

List Of Practical’s (Indicative & Not Limited To)

Sr. No.	Experiments	CO Mapping
1	Understanding Text Editors to Write Programs Compile and run first java file Byte Code and class file	**CO1**
2	Sketch a class and object diagram describing the sales order system of restaurant	**CO1**
3	Sketch a class diagram describing the circle and rectangle class	**CO1**
4	Sketch a class diagram for a college platform including, classroom, playground, chair, table, smart board, teaching staff etc.	**CO1**
5	Sketch a class diagram containing class called Employee, which models an employee with an ID, name and salary. Add method raiseSalary(percent) that increases the salary by the given percentage.	**CO1**
6	Program to display default value of all Primitive data types	**CO1**
7	Implement the code using main() method to calculate and print the Total and Average marks scored by a student from the input given through the command line arguments. Assume that four command line arguments name, marks1, marks2, marks3 will be passed to the main() method in the below class with name TotalAndAvgMarks.	**CO1**

8	Write code which uses if-then-else statement to check if a given account balance is greater or lesser than the minimum balance. Write a class BalanceCheck with public method checkBalance that takes one parameter balance of type double. Use if-then-else statement and print Balance is low if balance is less than 1000. Otherwise, print Sufficient balance.	**CO1**
9	A class NumberPalindrome with a public method isNumberPalindrome that takes one parameter number of type int. Write a code to check whether the given number is palindrome or not. For example Cmd Args: 333 333 is a palindrome	**CO1**
10	Write a class Fibonacci Series with a main method. The method receives one command line argument. Write a program to display fibonacci series i.e. 0 1 1 2 3 5 8 13 21	**CO1**
11	Write a Java Program to find the Factorial of a given number.	**CO1**
12	Java Program to create a class, methods and invoke them inside main method.	**CO1**
13	Write a Java program to illustrate the abstract class concept. Create an abstract class Shape, which contains an empty method numberOfSides(). Define three classes named Trapezoid, Triangle and Hexagon extends the class Shape, such that each one of the classes contains only the method numberOfSides(), that contains the number of sides in the given geometrical figure. Write a class AbstractExample with the main() method, declare an object to the class Shape, create instances of each class and call numberOfSides() methods of each class.	**CO1**
14	Java program to illustrate the static field in the class.	**CO1**
15	Java Program to illustrate static class.	**CO1**
16	Write a java program to access the class members using super keyword	**CO2**
17	Java program to access the class members using this keyword	**CO2**
18	Implement an interface named MountainParts that has a constant named TERRAIN that will store the String value “off_road”. The interface will define two methods that accept a String argument name newValue and two that will return the current value of an instance field. The methods are to be named: getSuspension, setSuspension, getType , setType.	**CO2**
19	Java program to demonstrate nested interface inside a interface.	**CO2**
20	Java program to demonstrate nested interface inside a class.	**CO2**
21	Java program to explicit implementation of garbage collection by using finalize() method	**CO2**
22	Java program to implement Single Inheritance	**CO2**
23	Java program to implement multi- level Inheritance	**CO2**
24	Java program to implement constructor and constructor overloading.	**CO2**
25	Java program implements method overloading.	**CO2**
26	Java program to implement method overriding.	**CO2**
27	Java program to implement lambda expression without parameter.	**CO2**
28	Java program to implement lambda expression with single parameter.	**CO2**
29	Java program to implement lambda expression with multi parameter.	**CO2**
30	Java program to implement lambda expression that iterate list of objects	**CO2**
31	Java program to define lambda expressions as method parameters	**CO2**
32	Write a class CountOfTwoNumbers with a **public** method compareCountOf that takes three parameters one is arr of type int[] and other two are arg1 and arg2 are of type int and returns true if count of arg1 is greater than arg2 in arr. The return type of compareCountOf should be boolean. Assummptions: • arr is never null	**CO2**

	arg1 and arg2 may be same	
33	Java program to show the multiplication of two matrices using arrays.	**CO2**
34	Java Program to search an element using Linear Search	**CO2**
35	Java program to search an element using Binary Search	**CO2**
36	Java Program to sort element using Insertion Sort	**CO2**
37	Java Program to sort element using Selection Sort– Largest element Method	**CO2**
38	Java program to Sort elements using Bubble Sort	**CO2**
39	Java program to create user defined package.	**CO3**
40	Java Program to create a sub- classing of package	**CO3**
41	Implement the following: 1. Import package.*; 2. import package.classname; Using fully qualified name.	**CO3**
42	Implement and demonstrate package names collision in java	**CO3**
43	Java program to handle and Arithmetic Exception Divided by zero	**CO3**
44	Java Program to implement User Defined Exception in Java	**CO3**
45	Java program to illustrate finally block	**CO3**
46	Java program to illustrate Multiple catch blocks	**CO3**
47	Java program for creation of illustrating throw in exception handling.	**CO3**
48	Implement the concept of Assertion in Java Programming Language	**CO3**
49	Implement the concept of Localization in Java Programming Language.	**CO3**
50	Java program to print the output by appending all the capital letters in the input string.	**CO3**
51	Java program that prints the duplicate characters from the string with its count.	**CO3**
52	Java program to check if two strings are anagrams of each other	**CO3**
53	Java Program to count the total number of characters in a string	**CO3**
54	Java Program to count the total number of punctuation characters exists in a String	**CO3**
55	Java Program to count the total number of vowels and consonants in a string	**CO3**
56	Java Program to show .equals method and == in java	**CO3**
57	Given a string, return a new string made of n copies of the first 2 chars of the original string where n is the length of the string. The string may be any length. If there are fewer than 2 chars, use whatever is there. If input is "Wipped" then output should be "WiWiWiWiWi".	**CO3**
58	Given two strings, a and b, create a bigger string made of the first char of a, the first char of b, the second char of a, the second char of b, and so on. Any leftover chars go at the end of the result. If the inputs are "Hello" and "World",then the output is "HWeolrllod".	**CO3**
59	Java program to show the usage of string builder.	**CO3**
60	Java program to show the usage of string buffer.	**CO3**
61	Creating and Running a Thread	**CO4**
62	Implementing Runnable Interface	**CO4**
63	Synchronizing Threads with lock	**CO4**

64	Synchronizing Threads without lock	**CO4**
65	Java program to implement even and odd threads by using Thread class .	**CO4**
66	Java program to implement even and odd threads by using Runnable interface.	**CO4**
67	Java program to synchronize the threads by using Synchronize statements and Synchronize block.	**CO4**
68	Write a program where the client sends a message to the server, and the server prints it by using TCP	**CO4**
69	Implement a server that can handle multiple clients simultaneously using UDP	**CO4**
70	Write a client-server application where the client uploads a file and the server saves it by using TCP/UDP.	**CO4**
71	Java program to implement that read a character stream from input file and print it into output file.	**CO4**
72	Java program to implement that merge the content of two files (file1.txt, file2.txt) into file3.txt.	**CO4**
73	Write a Java program that reads the contents of one file and copies them toanother file.	**CO4**
74	Write a Java program that reads a text file and counts the number of words in it.	**CO4**
75	Write a Java program that reads a text file and counts the frequency of each word in it.	**CO4**
76	Write a Java program that reads a text file and adds line numbers to each line. The program should create a new file with the line numbers added to the beginning of each line.	**CO4**
77	Write a Java program that reads two binary files and compares them byte by byte to determine if they are identical. Display a message indicating whether the files are the same or different.	**CO4**
78	Program to create a frame with three button in AWT and swing	**CO5**
79	Program to display message with radio buttons in swing	**CO5**
80	Program to display "All The Best" in 5 different colors on screen. (Using AWT/Swing)	**CO5**
81	Program to implement handling in a button “OK”	**CO5**
82	Java Program to implement BorderLayout	**CO5**
83	Java Program to implement GridLayout	**CO5**
84	Java Program to implement BoxLayout	**CO5**
85	Java Program to implement CardLayout	**CO5**
86	Java program to implement Generic class	**CO5**
87	Java program to illustrate Generic methods	**CO5**
88	Java program to implement wildcard in generics	**CO5**
89	Java program to implement of methods of HashSet	**CO5**
90	Java Program to implement methods available in HashMap class	**CO5**
91	Program to add, retrieve, and remove element from ArrayList	**CO5**
92	Create a method which can accept a collection of country names and add it to ArrayList with generic defined as String and return the List.	**CO5**
93	Create a method which can create a HashSet containing values 1-10.The Set should be declared with the generic type Integer.The method should return the Set.	**CO5**
94	Java program to implement autoboxing	**CO5**
95	Java program to implement unboxing	**CO5**
96	Develop a java class with a method *storeEvenNumbers(int N)* using ArrayList to store even numbers from 2 to N, where N is a integer which is passed as a parameter to the method *storeEvenNumbers().* The method should return the ArrayList (A1) created.	**CO5**
97	Create a method that accepts the names of five countries and loads them to an array list and returns the list.	**CO5**
98	Create a method which can accept a collection of country names and add it to ArrayList with generic defined as String and return the List.	**CO5**

Course Code: CNC0301/ CNC0401	Course Name: Artificial Intelligence and Cyber Ethics	L	T	P	C
Course Offered in: B. Tech. 2nd Year (3rd Semester/4th Semester) All Branches		2	0	0	-

Pre-requisite: Basic understanding of AI, Cybercrime, Computer System and Ethics.

Course Objectives:

The course aims to foster critical thinking about ethical issues, promote responsible use of technology, and ensure students can identify, analyze, and address ethical dilemmas in Artificial Intelligence and cyber domains.

Course Outcome: After completion of the course, the student will be able to		Bloom's Knowledge Level (KL)
CO1	Learn key principles of AI ethics, summarizing ethical considerations and applications in AI development and deployment.	K2
CO2	Apply policies and framework for Fairness in AI and Machine Learning.	K3
CO3	Apply privacy and security concepts, risk management and regulatory compliance in the field of AI and Cyber Security.	K3
CO4	Understand the nature of cybercrimes, the principles of intellectual property rights (IPR), and the legal measures necessary to address and prevent these issues.	K2
CO5	Describe the impact of AI in Society, employment and workforce.	K2

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	2	2	1	1	2	2	3	1	2	1	2	1	3	2
CO2	2	3	3	2	2	2	3	1	2	1	2	1	3	2
CO3	3	3	2	3	2	2	3	1	1	1	2	3	3	3
CO4	2	2	1	1	1	3	3	1	2	1	2	2	2	3
CO5	1	1	1	1	1	3	3	2	3	2	3	1	2	2

Course Contents / Syllabus

Module -1	An Overview to AI Ethics	05 hours

Definition of AI Ethical principles in AI, Sources of AI data, Legal implications of AI Security Breaches, Privacy and AI Regulations, Key Principles of Responsible AI, Transparency and Accountability, Dual-Use Dilemma, Human-Centric Design, Introduction to Cyber Laws and Ethics, Historical Development of Cyber laws, Legal frameworks

Module -2	Fairness and Favoritism in Machine Learning	06 hours

Introduction to Fairness and Bias in AI, Types of Fairness and Bias, Impact of Bias and Fairness in AI, Techniques for Measuring Fairness and Bias, Techniques for Mitigating Bias, Current Policies and Frameworks for Fairness in AI, Bias in Data Collection, Fairness in Data Processing, Generative AI, Types of Bias in Generative AI.

Module -3	AI Ethics and Cybersecurity Principles	05 hours

Importance of Privacy and Security in AI, AI specific Security Tools and Software, Privacy-Preserving Machine Learning (PPML) and Privacy-Preserving Data Mining (PPDM), Risk Management: Risk Assessment and Incident Response, Regulatory Compliance: GDPR, HIPAA, Case Studies: Implementation of AI Ethics guidelines and best practices in engineering projects.

Module -4	Cybercrimes, IPR and Legal Measures	08 hours

Types of Cybercrimes and their Impact, Legal measures for Cybercrime Prevention and Prosecution, IPR: Copyrights, Trademarks, Patents, and Trade Secrets, Ethical Implications of Intellectual Property, Cyber Security and Privacy Issues, Cyber Crime Investigations and Digital Evidence Handling, Overview of Indian Cyber Laws (IT Act 2000 and Amendments), Comparative Overview: Indian vs Global Cyber Laws, Case Study: The ATM Heist – Cosmos Bank Cyber Attack (India, 2018).

Module -5	AI Contribution to Social Evolution	06 hours

Positive and Negative Political impacts of AI, Role of AI in Social Media and Communication Platforms, AI-Generated Content and Deepfakes, Key Technical Stakeholders in AI Deployment: Developers, Researchers, Policymakers, Technical Impacts on Employment and Workforce Automation Technologies: Robotic

Process Automation (RPA), Autonomous Systems.**File and Disk Management:** File Access Mechanism, File Allocation Method, Disk Scheduling Algorithms

Modern Operating System: Virtual machines – hypervisor, Introduction to GPU. Introduction of CUDA Programming.

Total Lecture Hours	**30 hours**

Textbooks:

Sr. No.	Book	Author
1.	Artificial Intelligence: A Guide for Thinking Humans , Penguin Books, 2019.	Melanie Mitchell
2.	Cyber Ethics: Morality and Law in Cyberspace , 7th Edition (2023).	Richard Spinello, Jones & Bartlett Learning,

Reference books:

Sr. No.	Book	Author
1.	Artificial Intelligence and Ethics by , BPB Publications, 2023	S. B. Kishor, Debajit Biswas
2.	Cyber Security and Cyber Laws Cengage, India, 2022.	Alfred Basta, Nadine Basta, Sattwik Panda,

NPTEL/ YouTube/ Faculty Video Link:

1.	https://www.youtube.com/watch?v=VqFqWIqOB1g	6M51S
2.	https://www.youtube.com/watch?v=hVJqHgqF59A	5M19S
3	https://www.youtube.com/watch?v=O5RX_T4Tg24	44S
4	https://www.youtube.com/watch?v=RJZ0pxcZsSQ	5M01S
5	https://www.youtube.com/watch?v=I9FOswjTSGg	10M41S

Course Code: CNC0302/ CNC0402	Course Name: Environmental Science	L	T	P	C
Course Offered in: B. Tech. 2nd Year (3rd Semester/4th Semester) All Branches		2	0	0	-

Pre-requisite: Basic knowledge of biology, chemistry, ecology, geology, mathematics, and understanding of human impacts on natural systems.

Course Objectives:

Course Outcome: After completion of the course, the student will be able to		Bloom's Knowledge Level (KL)
CO1	Understand the basic principles of ecology and environment. Ecosystem: Basic concepts, components of ecosystem, food chains and food webs. Ecological pyramids, biodiversity.	K2
CO2	Understand the different types of natural recourses like food, forest, Minerals and energy and their conservation.	K2
CO3	Understand the different types of pollution, pollutants, their sources, effects and their control methods.	K2
CO4	Understand the basic concepts of sustainable development, Environmental Impact Assessment (EIA) and different acts related to environment	K2

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	2	2	-	3	3	2	2	-	2	1	1	1
CO2	3	3	2	2	-	3	3	2	2	-	2	1	1	1
CO3	3	3	2	2	-	3	3	2	2	-	2	1	1	1
CO4	3	3	2	2	-	3	3	3	2	-	2	2	1	2

Course Contents / Syllabus

Module -1	Basic Principle of Ecology and Biodiversity	05 hours

Definition, Scope and basic principles of ecology and environment. Ecosystem: Basic concepts, components of ecosystem. Food chains and food. Webs. Ecological pyramids, Energy flow in ecological systems, Characteristics of different ecosystems. Biogeochemical Cycles: Importance, gaseous and sedimentary cycles. Carbon, Nitrogen, Phosphorus and Sulphur Cycles. Biodiversity and their importance, Threats to biodiversity, major causes, extinction's, vulnerability of species to extinction, IUCN threat categories, Red data book. Strategies for biodiversity conservation, principles of biodiversity conservation in-situ and ex-situ conservation strategies Mega diversity zones and Hot spots, concepts, distribution and importance.

Module -2	Natural Resources and Ecological succession	06 hours

Natural resources and associated problems. Forest resources: Use and over- exploitation, deforestation. Timber extraction, mining, dams and their effects on forest and tribal people. Mineral resources: Use and exploitation, environmental effects of extracting and using mineral resources. Food resources: World food problems, changes caused by agriculture and over-grazing, effects of modern agriculture, fertilizer-pesticide problems, water logging, and salinity. Land resources: Land as a resource, land degradation, man induced landslides. Equitable use of resources for sustainable lifestyles.
Non-Renewable Energy Resources: Fossil fuels and their reserves, Nuclear energy, types, uses and effects, Renewable Energy Resources: hydropower, Solar energy, geothermal, tidal and wind energy, Biomass energy, biogas and its advantages. Ecological succession-Types, stages, examples of ecological succession

Module -3	Pollution and Waste Management	05 hours

Air pollution: sources of air pollution, Primary and secondary air pollutants. Origin and effects of SOX, NOX, Cox,CFC, Hydrocarbon, control of air pollution. Water pollution: sources and types of water pollution, Effects of water pollution, Eutrophication, Soil pollution: Causes of soil pollution, Effects of soil pollution, Major sources of and effects of noise pollution on health, Radioactive and thermal pollution sources and their effects on surrounding environment. Solid waste disposal and its effects on surrounding environment, Introduction to E- Waste, Types and classification of E-Waste, Impacts of E- Waste on environment and human health,E-Waste management and recycling., Climate change, global warming, acid rain, ozone layer depletion.

Module -4	Environmental Assessment and Legislation	08 hours

Women education, Role of NGOs regarding environmental protection, Bio indicators and their role, Natural disasters and disasters management, Aims and objectives of Environmental Impact Assessment (EIA). Salient features of following Acts: Environmental Protection Act, 1986, Wildlife (Protection) Act, 1972.Water (Prevention and control of pollution) Act, 1974. Forest (Conserving) Act, 1980.

Definition and concept of sustainability, impacted areas of sustainable development, Global initiative and issues on sustainable development UNSDsGs, System Thinking and Sustainability.

Total Lecture Hours	**30 hours**

Textbooks:

Sr. No.	Book	**Author**
1.	The nature and properties of Soils, Tenth Edition. Mac Millan Publishing Co., New York,1990.	Brady, N.C.
2.	Fundamentals of Environmental Chemistry: Narosa Publishing House, New Delhi.. 2005	Sodhi G.S
3	Fundamentals of Ecology, Tata Mc Graw Hill, New Delhi.. (1994),	Dash, M.C

Reference books:

Sr. No.	Book	**Author**
1.	Air Pollution, Tata McGraw Hill Publishing Co. Ltd., New Delhi, 1989	Rao M.N. and H.V.N. Rao,
2.	A Text Book of environmental Science	Shashi Chawla

NPTEL/ YouTube/ Faculty Video Link:

MOOCs	Mastering VMware vSphere 6.5	
Video Lectures	1.https://www.youtube.com/watch?v=T21OO0sBBfc, https://www.youtube.com/watch?v=qt8AMjKKPDo 2. https://www.youtube.com/watch?v=mOwyPENHhbc, https://www.youtube.com/watch?v=yqev1G2iy2 https://www.youtube.com/watch?v=_74S3z3lO_I, https://www.youtube.com/watch?v=jXVw6M6m2 3. https://www.youtube.com/watch?v=7qkaz8ChelI, https://www.youtube.com/watch?v=NuQE5fKmfME https://www.youtube.com/watch?v=9CpAjOVLHII, h t t p s ://www.youtube.com/watch?v=yEci6iDkXYw 4. https://www.youtube.com/watch?v=ad9KhgGw5iA, https://www.youtube.com/watch?v=nW5g83NSH9 M, https://www.youtube.com/watch?v=xqSZL4Ka8xo	7M42S 15M19S 31M35S 21M54S 13M46S 9M39S 5M08S 32M35S 24M29S

Course Code: CASCC0401	Course Name: Employability Skill Development - II	L	T	P	C
Course Offered in: IV Semester **Note:** Section C - offered in B.Tech. III/IV SEM		0	0	4	2

Pre-requisite: Students should have successfully completed communication skills courses in the first year.

Course Objectives:

Section A: Develop proficiency in C++ programming, data management, and project development by applying concepts, teamwork, and problem-solving to build practical applications.

Section B: Enhance quantitative aptitude skills through business mathematics concepts, enabling accurate problem-solving in financial calculations, ratios, and real-world numerical applications.

Section C: To develop the ability to communicate effectively and empathetically in workplace situations while strengthening active listening, adaptability, and cross-cultural awareness for navigating complex professional environments.

Course Outcome: After completion of the course, the student will be able to		Bloom's Knowledge Level (KL)
CO1	Apply C++ programming concepts, file handling, and STL techniques to develop efficient programs for data storage and manipulation.	K3
CO2	Design and implement a C++ -based project by applying planning, teamwork, and problem-solving skills.	K6
CO3	Solve real-life business math problems involving percentages, profit and loss, discounts, interest, averages, mixtures, and ratios using appropriate mathematical methods.	K3
CO4	Apply spoken English skills to participate effectively in standard workplace routines, including daily stand-ups, reporting, and collaborative problem-solving.	K3
CO5	Evaluate and adapt communication styles to suit dynamic workplace communication situations and deliver context-sensitive, and logically structured verbal responses under time constraints.	K6

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	-	2	3	1	-	3	2	2	2	-	-	-
CO2	3	3	3	3	3	2	1	3	2	2	2	-	-	-
CO3	3	3	2	1	-	-	-	1	2	2	2	-	-	-
CO4	-	-	-	-	1	2	-	3	2	2	-	-	-	-
CO5	-	2	1	-	1	2	-	3	2	2	1	-	-	-

Course Contents / Syllabus

Section – A

Module 1:	Foundations of C++ Programming and Data Management Techniques	08 hours

Core Fundamentals of C++, Functions & Program Structure, Arrays & Strings, File **Handling:** File streams (ifstream, ofstream, fstream), Reading/writing text files, saving user data (scores, records, logs) **STL Basics:** vector, array, map, unordered_map set Algorithms (sort, find)

Module 2:	Project/Game Development	08 hours

Project Planning & Development (Teams, roles, idea pitching and Implement in C++), Mini Project, Project Demonstration and Review

Section – B

Module 3	Business Math I	16 hours

Percentage, Profit and Loss, Discount, Simple Interest and Compound Interest, Average, Ratio & Proportion, Partnership, Mixture & Allegation, Clock , Calendar.

Section – C

Module 4	Workplace Interaction Essentials	12 hours

- The Professional Shift: Navigating the transition from academic to professional tone. Using appropriate register when addressing peers vs. supervisors.
- Active Listening & The Feedback Loop: Moving beyond passive hearing. Practicing the SBI (Situation-Behaviour-Impact) model for peer feedback and clarifying multi-step technical instructions.
- Structured Micro-Speaking: The art of the concise update. Delivering 1-2 minute "Stand-up" reports (What I did, What I will do, Blockers) and explaining basic technical processes to laymen.
- Guided Interpersonal Simulations: Role-playing everyday scenarios: politely interrupting a meeting, seeking assistance on a delayed task, and confirming understanding of complex briefs using paraphrasing techniques.
- Digital & Hybrid Nuances: Verbal signalling in virtual meetings, managing "overtalk," and translating concise written chat messages into spoken professional updates.

Module-5 **Workplace Communication in Action** **12 hours**

- Dynamic Problem-Based Communication: Simulating the "War Room." Explaining sudden technical failures, proposing immediate solutions, and negotiating revised deadlines with stakeholders.
- Listening Under Pressure & Global Adaptability: Interpreting fast, accented, or incomplete instructions (incorporating CEFR-level listening benchmarks). Decoding tone, urgency, and indirect cultural communication styles.
- Collaborative Conflict Resolution: Managing disagreements in team settings using "I-statements" and the "Yes, and…" technique to build consensus without escalating tension.
- The Persuasive Professional: Delivering evidence-based arguments using the PREP (Point, Reason, Example, Point) framework to justify a technical choice or resource allocation.
- Real-Time "Crisis" Simulation (Capstone): A comprehensive role-play where student groups are given a sudden project change or client issue. They must conduct an emergency meeting, delegate tasks verbally, and present a united response to the "client."

Total Lecture Hours **56 hours**

Reference Books:

Book Title	**Authors Name**
1. A Project-Based Introduction to Programming	Access Point Publishing
2. Programming: Principles and Practice Using C++	Bjarne Stroustrup
3. Effective Modern C++	Scott Meyers
4. Quantitative Aptitude	R S Aggarwal
5. Verbal & Non-Verbal Reasoning	R S Aggarwal
6. Crucial Conversations Tools	Kerry Patterson et al.
7. HBR Guide to Persuasive Presentations	Nancy Duarte
Self Study Content Link:	Duration(in hours)

MOOCs	nptel.ac.in/courses/109104107	8 weeks
Video Lectures	Employability skills - YouTube	33 m
Web Reference	Employability Skills Self-Management Guide | PDF | Goal | Self Esteem	-

Course Code: CCSE0401	Course Name: Database Management Systems	L	T	P	C
Course Offered in: CSE/CSE-R/IT/ M.Tech, Int./CSE (Twin)/IT (Twin)/CSE(Prof)/IT (Prof)/M&C/AI/AI (TWIN)/AIML/AIML (TWIN)/CS/CYS/DS/IOT		2	0	0	2

Pre-requisite: Basic understanding of computer fundamentals such as architecture, storage and hardware. In addition, familiarity with data structures, algorithms, and basic programming concepts will be beneficial.

Course Objectives:

The objective of the course is to introduce about database management systems, with an emphasis on how to organize, maintain and retrieve - efficiently, and effectively - information in relational & non-relational databases.

Course Outcome: After completion of the course, the student will be able to		Bloom's Knowledge Level (KL)
CO1	Apply ER model for conceptual design of the database.	K3
CO2	Execute SQL and apply the normalization to improve the database design.	K3
CO3	Implement complex queries in database with different applications.	K3
CO4	Understand transaction and concurrency control and execute the concept of PL/SQL.	K3
CO5	Analyze and implement effective Relational and Non-Relational database for real-world problems.	K4

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	3	3	2	1	1	1	2	1	2	3	1	2
CO2	3	2	3	3	2	2	1	2	1	2	2	3	2	2
CO3	3	3	2	3	3	2	-	2	1	-	2	3	3	2
CO4	3	2	2	2	2	2	1	-	1	1	1	3	3	2
CO5	2	2	2	2	3	2	1	-	1	2	2	3	3	2

Course Contents / Syllabus

Module 1	Introduction about the Database Conceptual Designing	08 hours

Basic Concept: Database system concept, architecture, History of Database, Data Independence, Database system Vs File system, Data models & Types of Data Models, schema and instances.

Data Modelling using the Entity Relationship Model: ER model concepts, Degree of relationship, Notation for ER diagram, mapping constraints, Generalization, Aggregation, Reduction of an ER diagrams to tables. Extended ER Diagram & reduction of EER. Codd Rules.

Introduction on SQL: Types of SQL commands: -DDL, DML, DCL, TCL. Basic of Relation Algebra & Operations, Query Optimization.

Module 2	Basic of SQL & Normalization	08 hours

Keys & Types of Keys: Super key, Candidate Key, Primary Key, Alternative Key, Foreign Key, unique. Constraints and Types of Constraints.

Use of Functions and Clause: Aggregate Functions, Scalar Functions, Where, Group by, Having and Order by.

Use of Predicates: SQL Operators. Like, Between, Aliases, distinct, limit.

Functional Dependencies, attribute Closure, Normalization & Types of Normalization, Candidate Key, FD Preservation, Lossy and Lossless decomposition, Canonical Cover of FD's, Denormalization.

Module 4	Transaction and Concurrency Control	08 hours

Transaction system: Life cycle of transaction, ACID Properties Schedule & Types of Schedule, Serializability, Recoverability, Deadlock Handling.

Concurrency Control Techniques: Concurrency Control, Concurrency control Techniques: Locking Techniques, Timestamping, Validation Based Protocol, Transaction & Data Control: -Grant, Revoke, commit & Rollback.

Module 5	Introduction of NoSQL With MongoDB	08 hours

Introduction to NoSQL: NoSQL Data Models, Overview of NoSQL Databases with their Types, Uses & Features of NoSQL Document Databases, CAP theorem, BASE Vs ACID, Comparison of relational databases to NoSQL stores, uses and deployment: MongoDB, Cassandra, HBASE, Neo4j, Redis and Riak.

Introduction to MongoDB: Introduction and Features of MongoDB, MongoDB Operators, MongoDB Collection & Document, MongoDB Shell & their commands, CRUD operations.

Introduction of Cloud Database. MongoDB Cloud product : Stitch, Atlas & Cloud Manager.

Total Lecture Hours	**32 hours**

Textbooks:

Book Title	**Authors Name**
1. Database Concepts, McGraw Hill, 7th Edition, 2020	Avi Silberschatz Henry F. Korth S. Sudarshan
2. Fundamentals of Database Systems, Addision Wesley, 7th edition, 2016	R. Elmasri and S. B. Navathe

Reference Books:

Book Title	**Authors Name**
1. Database Systems: A practical Approach to Design, Implementation and Management. Pearson Education,3rd Edition, 2007.	Thomas Connolly , Carolyn Begg
2. NoSQL and SQL Data Modeling: Bringing Together Data, Semantics, and Software, Ted Hills, 1st Edition,2016.	Ted Hills

Self-Study Content Links:		**Duration (in Hrs)**
MOOCs	nptel.ac.in/courses/106105175	8 Weeks
Video Lectures	DBMS L1 Inauguration & Introduction DBMS L2 Introduction to Relational Model DBMS L8C Entity Relationship Model DBMS L8D Entity Relationship Model (Problem Solving and Discussion) DBMS L4A Joins, Set Operations and Aggregate Functions DBMS L4B Joins, Set Operations and Aggregate Functions DBMS L9B Relational Database Design DBMS L9C Relational Database Design DBMS L5A Nested Subqueris DBMS L15 Transactions DBMS L16A Concurrency Control DBMS L16B Concurrency Control DBMS L16C Concurrency Control L64: Non-relational database management systems | nosql & cap theorem DBMS L19 Distributed Data Stores and NoSQL Databases	30 Hrs

Course Code: CMCT0401	**Course Name:** Quantum Information Theory	**L**	**T**	**P**	**C**
Course Offered in: Department of Mathematics and Computing, B.Tech. IV Sem		**3**	**0**	**0**	**3**

Pre-requisite: Introduction to Quantum Computing, Linear Algebra, Quantum Mechanics

Course Objectives:

This course introduces the core principles of quantum information theory, focusing on quantum bits, entanglement, and measurement. Students will explore classical and quantum Shannon theory, including data compression and channel capacity, while applying concepts like quantum entropy and mutual information to communication protocols. The course also covers the theoretical foundations of quantum cryptography and error correction.

Course Outcome: After completion of the course, the student will be able to		Bloom's Knowledge Level (KL)
CO1	Explain core concepts of quantum information such as qubits, superposition, and entanglement.	K2
CO2	Analyze classical and quantum information measures using entropy and mutual information.	K4
CO3	Demonstrate quantum communication protocols like superdense coding and teleportation.	K3
CO4	Evaluate the capacity of classical and quantum channels.	K5
CO5	Demonstrate the concept of quantum error correction and basic cryptographic protocols.	K3

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO1	3	3	2	2	2	-	-	-	1	-	-	2	2
CO2	3	3	3	3	3	-	-	-	1	-	-	2	2
CO3	3	3	3	3	3	-	-	-	1	-	-	2	3
CO4	3	3	3	3	3	-	-	-	1	-	-	2	3
CO5	3	3	3	3	3	-	-	-	1	-	-	2	3

Course Contents / Syllabus

Module 1	**Quantum Bits and Information**	**10 hours**

Introduction to classical bits and information; Qubits and Bloch sphere representation; Quantum state vectors and density matrices; Measurement theory and postulates of quantum mechanics

Module 2	**Quantum Entropy and Mutual Information**	**10 hours**

Shannon entropy and classical mutual information; Von Neumann entropy and quantum mutual information; Properties of entropy and relative entropy; Strong subadditivity and its implications

Module 3	**Quantum Communication Protocols**	**10 hours**

Quantum teleportation; Superdense coding; Quantum key distribution (BB84); Entanglement swapping

Module 4	**Channel Capacities and Data Compression**	**10 hours**

Classical channel capacity and coding theorem; Quantum channel capacity and Holevo bound; Schumacher compression; Entanglement-assisted communication

Module 5	**Quantum Error Correction and Cryptography**	**8 hours**

Basic concepts in quantum error correction; Shor code, Steane code; Introduction to quantum cryptography; No-cloning theorem and security implications

	Total Lecture Hours	**48 hours**

Textbook:

S.No	**Book Title**	**Author**

1	Quantum Computation and Quantum Information, Cambridge University Press. 10th edition, 2010	Michael A. Nielsen and Isaac L. Chuang
2	Quantum Information Theory, Cambridge University Press. 2nd edition, 2017	Mark M. Wilde

Reference Books:

S.No	**Book Title**	**Author**
1	Lecture Notes on Quantum Computation, CreateSpace Independent Publishing Platform. 1st edition, 2015.	John Preskill
2	Introduction to Quantum Information Science, Oxford university press, 1st edition, 2006.	Vedral, Vlatko

Self Learning /NPTEL/ Youtube/ Faculty Video Link:

Module 1	https://nptel.ac.in/courses/115104123	10
Module 2	https://www.youtube.com/playlist?list=PLUl4u3cNGP60uVBMaoNERc6knT_MgPKS0	10
Module 3	https://quantum.country/qcvc	08
Module 4	https://www.edx.org/course/quantum-information-science-i	08
Module 5	https://www.youtube.com/playlist?list=PL1826E60FD05B44E4	09

LAB Course Code:CMCT0451	**LAB Course Name:** Quantum Information Theory Lab	**L**	**T**	**P**	**C**
Course Offered in: Department of Mathematics and Computing, B.Tech. IV Sem		**0**	**0**	**2**	**1**

Pre-requisite: Python Programming, Basics of Quantum Mechanics and Quantum Computing

Course Objectives:

This course enables students to implement quantum information protocols using Python and Qiskit, exploring qubit operations, measurements, and gates through simulation. They will apply theoretical communication models practically, visualize concepts like entropy and mutual information, and design foundational circuits for secure communication and error correction.

Course Outcome: After completion of the course, the student will be able to		Bloom's Knowledge Level (KL)
CO1	Simulate qubit states and basic quantum operations using Python libraries.	K4
CO2	Implement quantum communication protocols such as teleportation and superdense coding.	K3
CO3	Demonstrate quantum key distribution and analyze measurement outcomes.	K4
CO4	Evaluate entropy and mutual information via practical experiments.	K5
CO5	Apply error correction and entanglement-based protocols in simulation	K3

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO1	3	3	3	3	3	-	-	-	2	-	-	2	3
CO2	3	3	3	3	3	-	-	-	2	-	-	2	3
CO3	3	3	3	3	3	-	-	-	2	-	-	2	3
CO4	3	3	3	3	3	-	-	-	2	-	-	2	3
CO5	3	3	3	3	3	-	-	-	2	-	-	2	3

List Of Practical's (Indicative & Not Limited To)	**CO Mapping**
1. Simulating single qubit states and visualizing them on the Bloch sphere.	CO1
2. Performing single and multi-qubit measurements and understanding their effects.	CO1
3. Implementing and visualizing quantum gates (X, H, Z, CNOT) using Qiskit.	CO1
4. Simulating quantum teleportation and analyzing its fidelity.	CO2
5. Implementing superdense coding protocol and decoding messages.	CO2
6. Simulating BB84 Quantum Key Distribution and analyzing measurement results.	CO3
7. Calculating and plotting quantum entropy and mutual information.	CO4
8. Visualizing effects of noise on quantum states and operations.	CO4
9. Demonstrating quantum error correction using 3-qubit bit-flip code.	CO5
10. Using density matrices to describe mixed states and partial tracing.	CO4
11. Entanglement swapping protocol simulation and fidelity analysis.	CO5
12. Mini project: Combine quantum protocols and simulate a secure quantum communication network.	All COs
	Total Hours: 30 hrs.

Course Code: CASMCT0401	Course Name: Applied Linear Algebra and Quantum Mechanics	L	T	P	C
Course Offered in: B. Tech. All branches (except CSBS)		3	0	0	2

Pre-requisite: Basic Linear Algebra, Classical Mechanics, Introduction to Quantum Computing.

Course Objectives:

- Understand the foundational concepts of linear algebra relevant to quantum mechanics.
- Apply vector spaces, eigenvalues/eigenvectors, and matrix decompositions in quantum computing.
- Develop understanding of quantum mechanics postulates using linear algebra.
- Model quantum systems and analyze quantum states using Dirac notation and operators.
- Bridge theoretical quantum mechanics with computational aspects relevant to quantum computing.

Course Outcome: After completion of the course, the student will be able to		Bloom's Knowledge Level (KL)
CO 1	Apply vector spaces, inner products, and matrix operations in the context of quantum mechanics.	K3
CO2	Analyze quantum systems using linear transformations, eigenvalues, and eigenvectors.	K3
CO3	Interpret and manipulate quantum states and operators using Dirac notation.	K4
CO4	Apply the principles of quantum measurement, unitary evolution, and quantum postulates.	K4
CO5	Connect mathematical formalism with physical quantum systems and their simulation on quantum computers.	K5

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	1	1	1	1	1	2	1	2	3	1	1	-	-	-
CO2	1	1	1	1	1	1	1	2	3	1	1	1	-	1
CO3	1	1	1	1	1	1	1	2	3	1	1	2	1	2
CO4	1	1	1	1	1	1	1	2	3	1	1	1	1	2
CO5	1	1	1	1	1	1	1	2	3	1	1	1	1	1

Course Contents / Syllabus

Module 1	Vector Spaces and Linear Transformations	9 Hours

Vector spaces, subspaces, basis, and dimension; Inner product spaces and orthogonality; Linear transformations and matrix representation; Change of basis and similarity transformations

Module 2	Matrix Theory and Eigen Concepts	9 Hours

Eigenvalues, eigenvectors, diagonalization; Spectral theorem for Hermitian matrices; Singular Value Decomposition (SVD); Applications to quantum state transformations

Module 3	Introduction to Quantum Mechanics	9 Hours

Postulates of quantum mechanics; Quantum states and wavefunctions; Operators, observables, and commutators; Dirac notation and Hilbert spaces

Module 4	Quantum Dynamics and Measurement	9 Hours

Unitary evolution and Schrödinger equation; Measurement postulate and projection; Quantum entanglement and density operators; No-cloning theorem and uncertainty principle

Module 5	Quantum Systems and Computation	9 Hours

Qubits and multi-qubit systems; Tensor product and entangled states; Quantum gates as unitary operations; Quantum teleportation and simple quantum algorithms

Total Lecture Hours	45 hours

Textbooks:

1. Linear Algebra and Its Applications, Pearson. 2. Introduction to Quantum Mechanics, Cambridge University Press. 3. Quantum Computation and Quantum Information, Cambridge University Press.	
Reference Books:	
Book Title	
Linear Algebra Done Right, Springer.	
Modern Quantum Mechanics, Pearson.	
Self-Study Content Links:	
Video Lectures	https://nptel.ac.in/courses/111104115 https://ocw.mit.edu/courses/18-06-linear-algebra-spring-2010/ https://nptel.ac.in/courses/115106065 https://ocw.mit.edu/courses/8-04-quantum-physics-i-spring-2016/ https://quantum.country/qcvc

Course Code: CCSDS0411	Course Name: Fundamentals of Cloud Computing	L	T	P	C
Course Offered in: Department Elective		3	0	0	3

Pre-requisite: Knowledge of basic computing units

Course Objectives:

To introduce students to the core concepts, models, and technologies of cloud computing, enabling foundational understanding of cloud-based services and infrastructure.

Course Outcome: After completion of the course, the student will be able to		Bloom's Knowledge Level (KL)
CO1	Explain cloud computing fundamentals, business benefits, cloud service models, deployment models, and AWS global infrastructure components such as Regions, Availability Zones, and Edge Locations.	K2
CO2	Apply AWS pricing models, billing tools, IAM concepts, security mechanisms, and shared responsibility principles for secure and cost-effective cloud usage.	K3
CO3	Configure and implement AWS networking and compute services including VPC, EC2, Route 53, Lambda, and Elastic Beanstalk for cloud-based application deployment.	K3
CO4	Analyze and compare AWS storage and database services such as S3, EBS, EFS, RDS, DynamoDB, Aurora, and Redshift based on application requirements and use cases.	K4
CO5	Design scalable, highly available, and optimized cloud architectures using AWS Well-Architected Framework, Auto Scaling, Elastic Load Balancing, and CloudWatch monitoring services.	K6

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	2	1	–	2	–	–	–	1	–	2	2	2	1
CO2	3	2	2	1	3	–	2	–	1	2	2	3	3	1
CO3	3	3	3	2	3	–	1	2	1	2	2	3	3	2
CO4	3	2	2	3	3	–	1	–	2	1	2	3	3	2
CO5	3	3	3	2	3	1	2	2	2	2	3	3	3	3

Course Contents / Syllabus

Unit - 1	Core Principles of Cloud Computing and AWS Fundamentals	09 hours

Business case for cloud adoption, advantages of cloud computing, overview of cloud computing concepts, AWS global infrastructure including Regions, Availability Zones, and Edge Locations. Cloud service models (IaaS, PaaS, SaaS) and deployment models (Public, Private, Hybrid). Introduction to AWS services and navigation of AWS Management Console.

Lab Work: AWS account setup and exploration of AWS Management Console. (Introductory by Faculty)

Unit - 2	Cloud Economics and Security	09 hours

Fundamentals of AWS pricing models, total cost of ownership (TCO), AWS billing and cost management tools, AWS Organizations and support plans. AWS Shared Responsibility Model, Identity and Access Management (IAM), role-based access control, account security, data protection, and compliance fundamentals.

Lab Work:

- Introduction to AWS IAM
- IAM user, roles, and policy configuration
- Billing dashboard and cost monitoring exploration

Unit - 3	Networking and Compute Services	08 hours

Fundamentals of cloud networking, Amazon Virtual Private Cloud (VPC), subnets, routing, security groups, and network ACLs. Domain Name System using Route 53 and content delivery using CloudFront. Compute services including Amazon EC2 (instance types, pricing models), introduction to containers, AWS Lambda (serverless computing), and Elastic Beanstalk for application

deployment.

Lab Work:

- Build your VPC and Launch a Web Server (Module 5)
- Introduction to Amazon EC2 (Module 6)
- AWS Lambda Implementation (Module 6)
- Activity: Elastic Beanstalk Deployment (Module 6)

Unit - 4	**Storage and Database Services**	**10 hours**

Overview of storage types including object, block, and file storage. Amazon S3 (buckets, storage classes, lifecycle policies), Amazon EBS, EFS, and Glacier. Database services including Amazon RDS, DynamoDB, Aurora, and Redshift along with their use cases and comparisons.

Lab Work:

- Working with EBS (Module 7)
- Build a Database Server (RDS) (Module 8)
- S3 bucket creation and storage management (console practice by Faculty)

Unit - 5	**Cloud Architecture, Monitoring, and Scaling**	**09 hours**

AWS Well-Architected Framework principles including operational excellence, security, reliability, performance efficiency, and cost optimization. Elastic Load Balancing, Auto Scaling, and high availability concepts. Monitoring and logging using Amazon CloudWatch, performance analysis, and AWS Trusted Advisor for optimization and best practices.

Lab Work:

- Scale and Load Balance your Architecture (Module 10)
- CloudWatch monitoring and performance analysis (console practice by Faculty)

	Total Lecture Hours	**45 hours**

Textbooks:

Book Title	**Authors Name**
1. Fundamentals of Cloud Computing, Nitya Publication, 2020	Dr. Arun Singh Chouhan, Bipin Pandey, Vishwas Srivastava
2. Cloud Computing Basics: A Non-Technical Introduction, Apress, 2022	Anders Lisdorf

Reference Books:

Book Title	**Authors Name**
1. Cloud Computing revised and updated edition, 2023	Nayan B. Ruparelia

Self-Study Content Links:		**Duration (in Hrs)**
MOOCs	1. AWS Academy Cloud Foundations 2. AWS Cloud Practitioner Essentials	40 – 50 Hrs 6 – 8 Hrs
Video Lectures	1. https://nptel.ac.in/courses/106/104/106104182/ 2. https://www.youtube.com/watch?v=M988_fsOSWo&t=4s 3. https://www.youtube.com/watch?v=JYq1AQkMdhE 4. https://www.youtube.com/watch?v=iSG_72VNBVs&t=55s 5. https://nptel.ac.in/courses/106/105/106105167/ 6. https://youtu.be/FZR0rG3HKIk?si=i9Ol3TdIeWtC-UUJ 7. https://aws.amazon.com/ 8. https://www.youtube.com/watch?v=36zducUX16w 9. https://www.youtube.com/watch?v=3WIJ4axzFlU 10. https://www.youtube.com/watch?v=m8iz4CFVWK0 11. https://www.youtube.com/watch?v=IKxigcbhsGk 12. https://www.youtube.com/watch?v=NbkPRn1mqlU 13. https://youtube.com/playlist?list=PL1TLTEHdRxDbFyipEb0KENRuBTI9yUu26&si=Si2LGUG6 fu6v0Jr3	41hrs

Course Code: CCSE0411	Course Name: PYTHON WEB DEVELOPMENT WITH DJANGO	L	T	P	C
Course Offered in: CSE/CSE(R)/ CSE(TWIN)/ IT/ IT(TWIN)/ M.Tech(Int.)		3	0	0	3

Pre-requisite: Students should have good knowledge of Python Programming and Python coding experience.

Course Objectives:

This course focuses on how to design and build statistics as well as dynamic webpages and interactive web-based applications. These courses mainly focus on how Python operates within web development using the increasingly popular Django framework.

Course Outcome: After completion of the course, the student will be able to		Bloom's Knowledge Level (KL)
CO1	Apply Python programming concepts and client-side technologies to develop Django-based web applications.	K3
CO2	Demonstrate the use of the Django web framework to design and implement dynamic web pages and interactive web-based applications.	K3
CO3	Analyze and implement account management and authentication mechanisms in Django applications.	K4
CO4	Analyze the role of database connectivity using SQLite in web application development and its impact on modern e-commerce and social platforms.	K4
CO5	Design, develop, and deploy a functional Django web application on a cloud platform by integrating web technologies, databases, and authentication services.	K6

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	2	2	3	1	3	-	1	3	-	3	3	3	3	3
CO2	2	2	3	1	3	-	1	-	-	2	3	3	3	3
CO3	2	2	2	2	2	-	-	2	-	2	2	2	2	1
CO4	2	1	2	1	2	-	-	1	1	2	2	3	3	1
CO5	2	1	3	2	3	-	-	3	2	3	3	2	2	2

Course Contents / Syllabus

Module 1	Python libraries for web development	09 hours

Collections-Container datatypes, Tkinter-GUI applications, Requests-HTTP requests, BeautifulSoup4-web scraping, Scrapy, Zappa, Dash, CherryPy, Turbo Gears, Flask, Web2Py, Bottle, Falcon, Cubic Web, Quixote, Pyramid

Module 2	Introduction to Django Framework	09 hours

Understanding Django environment, Features of Django and Django architecture, MVC and MTV, Urls and Views, Mapping the views to URLs, Django Template, Template inheritance Django Models, Creating model for site, Converting the model into a table, Fields in Models, Integrating Bootstrap into Django, Creating tables, Creating grids,
Creating carousels.

Module 3	Integrating Accounts & Authentication on Django	09 hours

Introduction to Django Authentication System, Security Problem & Solution with Django Creating Registration Form using Django, Adding Email Field in Forms, Configuring email settings, Sending emails with Django, Adding Grid Layout On Registration Page, Adding Page Restrictions, Login Functionality Test and Logout.

Module 4	Connecting SQLite with Django	09 hours

Database Migrations, Fetch Data From Database, Displaying Data On Templates, Adding Condition On Data, Sending data from url to view, Sending data from view to template, Saving objects into database, Sorting objects, Filtering objects, Deleting objects, Difference between session and cookie, Creating sessions and cookies in Django.

Module 5	Deploying Django Web Application on Cloud	9 hours

Creating a functional website in Django, Four Important Pillars to Deploy, registering on Heroku and GitHub, Push project from Local System to GitHub, Working with Django Heroku, Working with Static Root, Handling WSGI with gunicorn, Setting up Database

& adding users.

Total Lecture Hours	**45 hours**

Textbooks:

	Authors Name
1."Python: The Complete Reference Paperback", 4th Edition 2018, McGraw Hill Education Publication.	Martin C. Brown
2."Python Programming: Using Problem Solving Approach", 3rd Edition 2017, Oxford University Press Publication.	Reema Thareja
3." Beginning Django Web Application Development and Deployment with Python", 2nd Edition 2017, Apress Publication.	Daniel Rubio, Apress
4."Python Django Web Development: The Ultimate Django web framework guide for Beginners", 2nd Edition 2019, Kindl e Edition.	William Jordon,

Reference Books:

Book Title	**Authors Name**
1."Building Django 2.0 Web Applications: Create enterprise-grade, scalable Python web applications easily with Django 2.0", 2nd Edition 2018, and Packt Publishing.	Tom Aratyn,
2."Build a website with Django", 1st Edition 2019, GNW Independent Publishing Edition.	Nigel George
3." Django in 8 Hours: For Beginners, Learn Coding Fast! 2nd Edition 2020, independently published Edition.	Ray Yao
4."Test-Driven Development with Python: Obey the Testing Goat: Using Django, Selenium, and JavaScript", 2nd Edition 2019, Kindle Edition.	Harry Percival

Self-Study Content Links:		**Duration (in Hrs)**
Video Lectures	https://youtu.be/eoPsX7MKfe8?list=PLIdgECt554OVFKXRpo_kuI0XpUQKk0ycO%20 https://youtu.be/tA42nHmmEKw?list=PLh2mXjKcTPSACrQxPM2_1Ojus5HX88ht7%20 https://youtu.be/8ndsDXohLMQ?list=PLDsnL5pk7-N_9oy2RN4A65Z-PEnvtc7rf%20 https://youtu.be/QXeEoD0pB3E?list=PLsyeobzWxl7poL9JTVyndKe62ieoN-MZ3%20 https://youtu.be/9MmC_uGjBsM?list=PL3pGy4HtqwD02GVgM96-V0sq4_DSinqvf https://youtu.be/F5mRW0jo-U4%20 https://youtu.be/yD0_1DPmfKM?list=PLQVvvaa0QuDe9nqlirjacLkBYdgc2inh3%20 https://youtu.be/rHux0gMZ3Eg%20 https://youtu.be/jBzwzrDvZ18%20https://youtu.be/RiMRJMbLZmg https://youtu.be/8DF1zJA7cfc%20 https://youtu.be/CTrVDi3tt8o%20 https://youtu.be/FzGTpnI5tpo%20 https://youtu.be/z4lfVsb_7MA https://youtu.be/WuyKxdLcw3whttps://youtu.be/WuyKxdLcw3w https://youtu.be/UxTwFMZ4r5k%20 https://youtu.be/2Oe55iXjZQI%20 https://youtu.be/zV8GOI5Zd6E%20 https://youtu.be/uf2tdzh7Bq4%20 https://youtu.be/RzkVbz7Ie44 https://youtu.be/kBwhtEIXGII%20 https://youtu.be/Q_YOYNiSVDY%20 https://youtu.be/_3AKAdHUY1M%20 https://youtu.be/6DI_7Zja8Zc%20 https://youtu.be/UkokhawLKDU	42 hrs

Course Code: CASL0401	Course Name: Technical Communication	L	T	P	C
Course Offered in: B. Tech. All branches (except CSBS)		2	0	0	2

Pre-requisite: Intermediate level (CEFR) and above.

Course Objectives:

Demonstrate effective verbal and non-verbal communication skills in diverse professional settings, including meetings, presentations, and interpersonal interactions.Develop and apply clear, concise, and audience-appropriate written communication, such as emails, letters, memos, resume', using correct grammar, tone, and format. Adapt communication style based on cultural, organizational, and situational contexts to foster inclusive and respectful professional relationships. Employ digital communication tools and platforms (e.g., video conferencing, business messaging apps) responsibly and effectively in remote or hybrid work environments.

Course Outcome: After completion of the course, the student will be able to		Bloom's Knowledge Level (KL)
CO1	Comprehend the principles and functions of technical communication.	K2
CO2	Write for specific audience and purpose to fulfil the provided brief	K3
CO3	Recognize and produce different kinds of technical documents.	K3
CO4	Apply effective speaking skills to efficiently carry out official discourses.	K3
CO5	Demonstrate their understanding of communication through digital media.	K3

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	1	1	1	1	1	2	1	2	3	1	1	-	-	-
CO2	1	1	1	1	1	1	1	2	3	1	1	1	-	1
CO3	1	1	1	1	1	1	1	2	3	1	1	2	1	2
CO4	1	1	1	1	1	1	1	2	3	1	1	1	1	2
CO5	1	1	1	1	1	1	1	2	3	1	1	1	1	1

Course Contents / Syllabus

Module 1	Introduction to Technical Communication	4 Hours

Technical Communication: Definition, Process, Types, Levels, and Flow; **Barriers to Technical Communication:** emphasis on gender neutral language and cultural sensitivity; **Significance of audience in technical communication**

Module 2	Technical Writing 1	5 Hours

Technical writing skill: characteristics, examples; **Business letters/emails**: Content organization, Tone and intent; **Agenda & Minutes of Meetings**

Module 3	Technical Writing 2	5 Hours

Job application, Resume'; Report, proposal; Technical paper: Abstract; Ethical Writing: Copy Editing, Referencing and Plagiarism

Module 4	Public Speaking	6 Hours

Job application, Resume'; Report, proposal; Technical paper: Abstract; Ethical Writing: Copy Editing, Referencing and Plagiarism

Module 5	Virtual/Remote Communication	4 Hours

Remote work: online platforms; **Video conferencing**; **Virtual etiquette**: email ids, usernames; **Writing Blogs & creating Vlogs**

Total Lecture Hours	24 hours

Textbooks:

	Authors Name
1.Technical Communication – Principles and Practices, 4th Edition , Oxford Univ. Press, 2022, New Delhi.	Meenakshi Raman & Sangeeta Sharma

Reference Books:

Book Title	Authors Name
1.Technical Communication, 15th Edition , Pearson, 2021	John M. Lannon & Laura J. Gurak
2.Spoken English- A Manual of Speech and Phonetics (5th Edition) by , 2024, New Delhi.	R K Bansal & J B Harrison, Orient Blackswan

3.Business Correspondence and Report Writing (6 Edition), Tata McGraw Hill & Co. Ltd., 2020, New Delhi.	Prof. R C Sharma, Krishna Mohan, and Virendra Singh Nirban
4.Intercultural Communication in Virtual Exchange , Cambridge Univ. Press, 2024.	Francesca Helm
Self-Study Content Links:	
Video Lectures	https://onlinecourses.nptel.ac.in/noc24_ge37/preview https://archive.nptel.ac.in/courses/109/106/109106094/ https://www.youtube.com/watch?v=kOJlwMJxEG0&t=8s https://www.youtube.com/watch?v=Sg7Q_dC_fWU&list=PLPuC5CMHiqmuzq_KQ4aw0V9Q7xJY6aezb https://www.youtube.com/watch?v=ymLFJDpjgCk&list=PLPuC5CMHiqmuzq_KQ4aw0V9Q7xJY6aezb&index=6

Course Code: CCSE0451	**LAB Course Name: Database Management Systems Lab**	**L**	**T**	**P**	**C**
Course Offered in: CSE/CSE-R/IT/ M.Tech, Int./CSE (Twin)/IT (Twin)/CSE(Prof)/IT (Prof)/ M&C/AI/AI (TWIN)/AIML/AIML (TWIN)/CS/CYS/DS/IOT		**0**	**0**	**4**	**2**

Pre-requisite: Basic knowledge of computer fundamentals, programming, data structures, relational database concepts.

Course Objectives:

To familiarize the students to the basics of database design and implementation.

Course Outcome: After completion of the course, the student will be able to		**Bloom's Knowledge Level (KL)**
CO1	Design and implement ER/EER models to solve real-world problems and transform them into relational schemas using appropriate database tools.	K3
CO2	Apply SQL and PL/SQL to create complex data queries, and procedural operations including triggers and functions, along with database connectivity.	K3
CO3	Analyze and ensure database integrity using constraints, and implement unstructured databases using MongoDB with appropriate query operations.	K4

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)

CO-PO Mapping	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PSO1**	**PSO2**	**PSO3**
CO1	3	3	3	3	2	1	-	1	2	1	2	3	1	2
CO2	3	3	3	3	2	2	-	2	1	2	2	3	3	2
CO3	2	2	2	2	3	2	-	-	1	2	2	3	3	2

List Of Practical's (Indicative & Not Limited To)

S. No.	**Name of Experiments/ Programs**	**CO**
1	Understand and implement the different ER diagram notation with their relationship and cardinalities.	**CO1**
2	Choose a case study on different domains I. E-commerce Platform II. Inventory Management III. Railway System IV. Hospital Data Management V. Company System VI. Banking System	**CO1**
3	Create an ER diagram for the above chosen case-study (Program-2). Identify the suitable entities and find the relationship and cardinalities between the entities with their relevant attributes.	**CO1**
4	Implement DDL, DML, DCL & TCL commands.	**CO1**
5	Implementation of I/O Constraints: Primary Key, Composite primary Key, Foreign Key, Unique Key.	**CO2**
6	Implementation of Business Constraints: Null, Not Null, Default, Check.	**CO2**
7	Practicing Queries using Like, Between, Aliases, distinct Operator & Predicate and Implement Aggregate functions.	**CO2**
8	Implementation of Queries using Where, Group by, Having and Order by Clause.	**CO2**
9	Create a table EMPLOYEE with following schema:-(Emp_no, E_name, E_address, E_ph_no, Dept_no, Dept_name,Job_id, Designation, Salary) Write SQL statements for the following query. i. List the E_no, E name, Salary of all employees working for MANAGER. ii. Display all the details of the employee whose salary is more than the Sal of any IT PROFF.	**CO2**

	iii. List the employees in the ascending order of Designations of those joined after 1981. iv. List the employees along with their Experience and daily. v. List the employee who are either 'CLERK' or 'ANALYST'. vi. List the employees who joined on 1-MAY-81, 3-DEC-81, 17-DEC-81. vii. List the e_name those are starting with 'S'. viii. Display total salary spent for each job category. ix. Display lowest paid employee details under each manager. x. Display number of employees working in each department and their department name. xi. Display the details of employees sorting the salary in increasing order. xii. Show the record of employee earning salary greater than 16000 in each department. xiii. Add constraints to check, while entering the empno value (i.e) empno> 100. xiv. Define the field DEPTNO as unique.Create a primary key constraint for the colum(EMPNO).	
10	Implementation of Queries using set theory operators UNION, INTERSECT, MINUS.	**CO2**
11	Implementation of Queries using Inner Join and Outer Join along with their types.	**CO2**
12	Implementation of Queries ,nested Queries or Sub Queries: IN, NOT IN, Exists, Not Exists, All and Any.	**CO2**
13	Apply the set theory operators, joins and nested queries on the chosen above case-study(Program-2)	**CO2**
14	Understand & implement the Database Connectivity with any programming language.	**CO2**
15	Implementation of Indexing, Views and sequence.	**CO2**
16	I. Write a PL/SQL Program to Add Two Numbers II. Write PL/SQL Program for Fibonacci Series III. Write PL/SQL Program to Find Greatest of Three Numbers	**CO2**
17	Write a PL/SQL code block to calculate the area of a circle for a value of radius varying from 3 to 7. Store the radius and the corresponding values of calculated area in an empty table named Areas, consisting of two columns Radius and Area.	**CO2**
18	Write a PL/SQL code block that will accept an account number from the user, check if the users balance is less than the minimum balance, only then deduct Rs.100/- from the balance.	**CO2**
19	Create an implicit cursor for the customers table to update the salary of all customers whose age is less than or equal to 25 by ₹500 and determine whether any rows are affected or not.	**CO2**
20	Create a row level trigger for the customers table that would fire for INSERT or UPDATE or DELETE operations performed on the CUSTOMERS table. This trigger will display the salary difference between the old values and new values.	**CO2**
21	Implementation of commit and rollback statement with amount transfer example.	**CO2**
22	Study of Open Source NOSQL Database and installation of MongoDB.	**CO3**
23	Implementation of the MongoDB Shell commands.	**CO3**
24	Implementation of the CRUD Operations in MongoDB.	**CO3**
25	Implementation of Aggregate Functions in MongoDB.	**CO3**
		Total Hours: 48 hrs.

Course Code: CCSE0455	Course Name: Web Technologies	L	T	P	C
Course Offered in: CSE/CS/IT/CSE(AI)/CSE(AIML)/CSE(IOT)/CSE(AI)/CSE(DS)		0	0	6	3

Pre-requisite:

1. Basic Programming Knowledge
2. Knowledge of any programming language (e.g., C, C++, Python/Java)
3. Familiarity with basic concepts of Internet.

Course Objectives:

This course covers different aspect of web technology such as HTML, CSS, Java Script and provide fundamental concepts of Internet, Web Technology and Web Programming. Students will be able to build a proper responsive website.

Course Outcome: After completion of the course, the student will be able to		Bloom's Knowledge Level (KL)
CO1	Identify the basic facts and explaining the basic ideas of Web technology and internet.	K2
CO2	Applying and creating various HTML5 semantic elements and application with working on HTML forms for user input.	K6
CO3	Understanding and applying the concepts of Creating Style Sheet CSS3 and bootstrap.	K3
CO4	Analysing and implementing concept of JavaScript and its applications.	K6
CO5	Creating and evaluating dynamic web pages using the concept of PHP.	K6

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	2	-	2	-	-	-	2	2	-	3	1	1
CO2	3	2	3	2	3	-	-	-	2	1	2	1	2	3
CO3	3	2	3	-	3	-	-	-	2	2	2	2	3	2
CO4	3	3	3	2	3	-	1	-	2	2	2	3	2	3
CO5	3	3	3	2	3	-	-	-	2	2	2	2	1	2

Course Contents / Syllabus

Unit 1	Basics of Web Technology & Testing	16 hours

Introduction: Introduction to Web Technology, History of Web and Internet, Connecting to Internet, Introduction to Internet services and tools, Client- Server Computing, Protocols Governing Web, Basic principles involved in developing a web site, Planning process, Types of Websites, Web Standards and W3C recommendations.

Web Hosting: Web Hosting Basics, Types of Hosting Packages, Registering domains, Defining Name Servers, Using Control Panel, Creating Emails in Cpanel, Using FTP Client, Maintaining a Website.

Unit 2	Introduction to HTML & XML	16 hours

HTML: What is HTML, DOM- Introduction to Document Object Model, Basic structure of an HTML document, Mark up Tags, Heading-Paragraphs, Line Breaks, Understand the structure of HTML tables. Lists, Working with Hyperlinks, Image Handling, Understanding Frames and their needs, HTML forms for User inputs. New form Elements- date, number, range, email, search and data list, Understanding audio, video and article tags.

XML: Introduction, Tree, Syntax, Elements, Attributes, Namespaces, Display, HTTP request, Parser, DOM, XPath, XSLT, XQuerry, XLink, Validator, DTD, Schema, Server.

Unit 3	Concepts of CSS3 & Bootstrap	16 hours

Concept of CSS 3: Creating Style Sheet, CSS Properties , CSS Styling(Background, Text Format, Controlling Fonts) , Working with block elements and objects , Working with Lists and Tables , CSS Id and Class, Box Model(Introduction, Border properties, Padding Properties, Margin properties) CSS Advanced(Grouping, Dimension, Display, Positioning, Floating, Align, Pseudo class, Navigation Bar, Image Sprites, Attribute sector) , CSS Color, Creating page Layout and Site.

Bootstrap: Introduction, Bootstrap grid system, Bootstrap Components.

Unit 4	JavaScript and ES6	16 hours

JavaScript Essentials: Introduction to Java Script , Javascript Types , Var, Let and Const Keywords, Operators in JS , Conditions Statements , Java Script Loops, JS Popup Boxes , JS Events , JS Arrays, Working with Arrays, JS Objects ,JS Functions , Using Java Script in Real time , Validation of Forms, Arrow functions and default arguments, Template Strings, Strings methods, Callback

functions, Object de-structuring, Spread and Rest Operator, Typescript fundamentals, Typescript OOPs- Classes, Interfaces, Constructor etc. Decorator and Spread Operator Difference == & ===, Asynchronous Programming in ES6, Promise Constructor, Promise with Chain, Promise Race.

Unit 5	**JavaScript and ES6**	**16 hours**

Introduction to PHP, Basic Syntax, Variables & Constants, Data Type, Operator & Expressions, Control flow and Decision making statements, Functions, Strings, Arrays. Working with files and directories: Understanding file& directory, Opening and closing, a file, Coping, renaming and deleting a file, working with directories, Creating and deleting folder, File Uploading & Downloading.

Session & Cookies: Introduction to Session Control, Session Functionality What is a Cookie, Setting Cookies with PHP. Using Cookies with Sessions, Deleting Cookies, Registering Session variables, Destroying the variables and Session.

	Total Lecture Hours	**80 hours**

Textbooks:

	Authors Name
1.“Web Technology and Design”, 1nd Edition 2003, New Age International.	C Xavier
2.“Internet and Web Technologies”, 2nd Edition 2017,Mc Graw Hill Education.	Raj Kamal,
3.“Beginning PHP Laravel”,2nd Edition 2020, kindle Publication.	Oluwafemi Alofe

Reference Books:

Book Title	**Authors Name**
1.“Collaborative Web Development” 5th Edition 1999, Addison Wesley Publication.	Burdman, Jessica
2.“Fundamentals of Web Development”,3rd Edition 2016	Randy Connolly
3.cHTML, DHTML, Java Script, Perl & CGI”, 4th Edition 2010 BPB Publication	Ivan Bayross,

Self-Study Content Links:		**Duration (in Hrs)**
Video Lectures	https://youtu.be/96xF9phMsWA%20	17 m 50 s
	https://youtu.be/Zopo5C79m2k%20	8 m 49s
	https://youtu.be/ZliIs7jHi1s%20	17 m 30 s
	https://youtu.be/htbY9-yggB0	4 m 35s
	https://youtu.be/vHmUVQKXlVo%20	7 m 55 s
	https://youtu.be/qz0aGYrrlhU%20	1 h 9 m 33 s
	https://youtu.be/BsDoLVMnmZs%20	3 h 11 m 11 s
	https://youtu.be/a8W952NBZUE	4 h 08 m 09 s
	https://youtu.be/1Rs2ND1ryYc%20	6 h 18 m 37 s
	https://youtu.be/vpAJ0s5S2t0%20	2 h 09 s
	https://youtu.be/GBOK1-nvdU4%20	3 h 18 m 17 s
	https://youtu.be/Eu7G0jV0ImY	8 h 39 m 53s
	https://youtu.be/-qfEOE4vtxE%20	2 h 46 m 21 s
	https://youtu.be/PkZNo7MFNFg%20	3 h 36 m 42 s
	https://youtu.be/W6NZfCO5SIk%20	48 m 16 s
	https://youtu.be/DqaTKBU9TZk	2 h 31 m 19s
	https://youtu.be/_GMEqhUyyFM%20	4h 25 m 04 s
	https://youtu.be/ImtZ5yENzgE%20	7 h 34 m 13 s
	https://youtu.be/xIApzP4mWyA%20	7 h 41 m 03 s
	https://youtu.be/qKR5V9rdht0	

Course Code: CCSCC0452	Course Name: Problem Solving Approaches	L	T	P	C
Course Offered in: CSE (DS)		0	0	2	1

Pre-requisite: Programming Language C/C++ or Java or Python

Course Objectives:

Problem-solving in computer programming involves a structured approach to identifying, analyzing, and resolving coding challenges. The process typically includes thoroughly understanding the problem, decomposing it into smaller, manageable parts, designing an appropriate algorithm, implementing the solution through code, and performing testing and debugging to ensure correctness and efficiency

Course Outcome: After completion of the course, the student will be able to		**Bloom's Knowledge Level (KL)**
CO1	Develop logic-based solutions using control statements, recursion and bit manipulation to solve basic and intermediate computational problems.	K6
CO2	Implement and manipulate arrays and strings using fundamental and advanced searching sorting techniques.	K3
CO3	Analyze and debug code for logical errors and improve the efficiency of the solution using appropriate data structures and algorithmic patterns.	K4

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	3	2	2	1	2	2	-	-	2	3	3	2
CO2	3	3	2	2	2	-	2	-	-	-	2	2	2	2
CO3	3	3	2	2	3	1	2	2	-	-	3	3	3	2

List Of Practical's (Indicative & Not Limited To)

Problem Statements need to be discussed in lab session: Control Statements

1. **Secure Password Generator**

A company wants to create a secure password generator for their employees. The password must be based on specific numeric properties to enhance its complexity and security. Write a program to validate and generate a secure password according to the following rules:

1. Prime Number Validation:

- The user must input a 3-digit number. The program should first check if the number is a prime number.
- If it is not a prime number, the user should be prompted to enter another number until a valid prime number is provided.

2. Sum of Digits Check:

- Once a valid prime number is entered, calculate the sum of its digits. If the sum of the digits is not divisible by 3, ask the user to enter another prime number until a valid one is found.

3. Armstrong Number Check:

- Check entered prime number is Armstrong or not? If Armstrong are found, prompt the user to enter another prime number and repeat the process.

Password Generation:

Concatenate the 1 if entered prime number is Armstrong otherwise 2 with the sum of the digits of the valid prime number to form the secure password.

Example Scenario:

Sample Input

Enter a 3-digit prime number: 153 Sum of digits of 153 = 9

The sum is divisible by 3. 153 is Armstrong number *Sample Output*

Secure Password: 19

2. Write a function to input electricity unit charges and calculate total electricity bill according to the given condition: For first 50 units Rs. 0.50/unit
For next 100 units Rs. 0.75/unit For next 100 units Rs. 1.20/unit For unit above 250 Rs. 1.50/unit
An additional surcharge of 20% is added to the bill

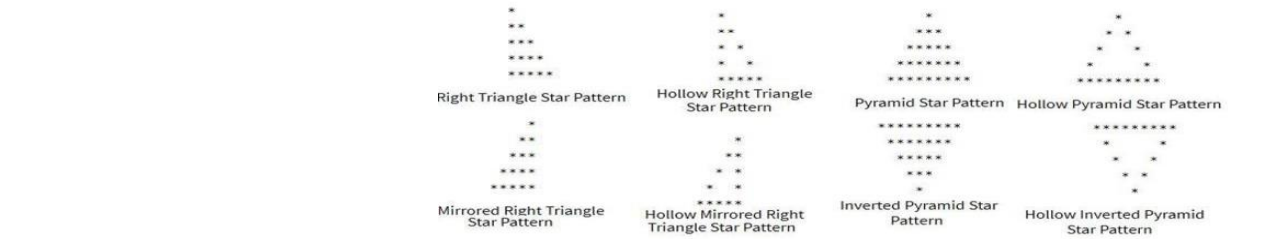
2. Write a method to generate a secure code which the sum of all possible palindrome numbers between given two numbers. For Example:

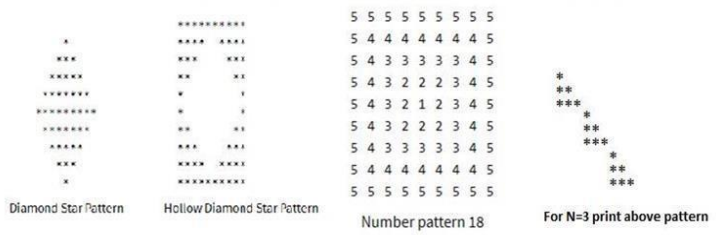
Input: 10, 80

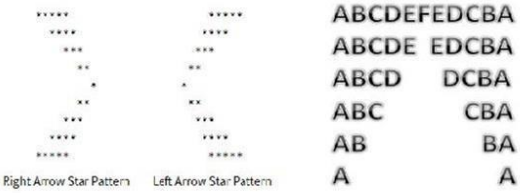
Output: 308

Explanation: All palindrome numbers between 10 & 80 are: 11,22,33,44,55,66,77 Password= 11+22+33+44+55+66+77 = 308

3. Draw **the following Patterns for N=5**







Problem Statements need to be discussed in lab session: Recursive Approach (Basic)

1. Write a program that takes an integer n as input and prints the multiplication table of n from n * 1 to n * 10. The output should clearly show each multiplication step.

2. Write a program to calculate the sum of all integers from 1 to a given number N. The program should take N as input and output the total sum using iteration or recursion.

3. Find the GCD of Two Numbers Using Recursion:

Write a recursive function to calculate the Greatest Common Divisor (GCD) of two numbers using Euclid's algorithm. The function should take two integers as input and return their GCD.

4. Find the LCM of Two Numbers Using Recursion:

Write a program to compute the Least Common Multiple (LCM) of two numbers using recursion. You may use the relationship LCM(a, b) = |a
* b| / GCD(a, b) and a recursive function for GCD.

Problem Statements need to be discussed in lab session: Bit Manipulation

1. Write a program to count the number of set bits (1s) in the binary representation of a given integer. The program should efficiently use bitwise operations to perform the task without converting the number to a string.

2. Write a program that takes a number and a bit position as input and checks whether the bit at that position is set (1) or clear (0). Use bitwise operators to perform the check

3. Given a number and a position, write a program to toggle (invert) the bit at the given position using bitwise operations. The result should reflect the updated value of the number after flipping the bit.

4. Write a program to compute the XOR of all numbers from 1 to n using a mathematical pattern (not a loop). Use bitwise XOR properties to achieve an efficient solution.

5. Given an array of size n-1 containing unique elements from 1 to n, find the missing number using bit manipulation (preferably XOR approach) without sorting or using extra space.

6. Given an array where all elements repeat twice except two elements that appear only once, write a program to find the two non-repeating elements using bitwise operations in linear time and constant space.

7. Write a program to check if a given number is a power of two using bit manipulation. A number is a power of two if it has exactly one set bit in its binary representation.

8. Given two integers A and B, write a program to count how many bits need to be flipped to convert A to B. Use XOR to find differing bits and count the number of set bits.

9. Write an efficient program to count the total number of set bits in binary representations of all numbers from 1 to n. Optimize the approach using bitwise logic and recursion.

10. Write a program to calculate the square of a number using only bitwise operations and addition. Do not use multiplication, division, or any power functions.

11. Write a function to add two integers using bitwise operations only. Avoid using the + or - operators. Implement logic using XOR and AND operations for binary addition.

12. Write a program to generate the power set (all subsets) of a given set using bitwise representation. Each subset can be represented by a binary number where each bit indicates inclusion of the corresponding element.

Problem Statements need to be discussed in lab session: Arrays (Try to use sliding window, prefix sum, cadence, recursion, bit manipulation, two pointer approaches)

1. Sarah is assisting the "MathMinds Club" in creating passwords for their online platform. They have a list of numbers, some stable and some unstable. Define a function that can help Sarah calculate the password according to the given scenario.

Scenario:

- There are N numbers provided.
- A number is stable if each digit appears the same number of times.
- A number is unstable if the frequency of its digits is not the same.
- The password is computed as the sum of all stable numbers minus the sum of all unstable numbers.
- Consider only those numbers in the list that have more than equal to three digits.

For example:

Input: N=5 List: 12, 1313, 122, 678, 898

Output: Password: 971

2. Given an array of integers, including possible negative values, you are allowed to modify at most one element by doubling its value. The goal is to find the maximum possible sum of any subarray after making this modification.

Input:

arr = [-2, 1, -3, 4, -1, 2, 1, -5, 4]

Expected Output:

- Original Maximum Subarray Sum: 6 (achieved from [4, -1, 2, 1])

Maximum Sum After Modification: 10(achieved from [8, -1, 2, 1], where the value 4 is doubled to 8).

3. For a given string, generate a pattern based on the following rules:

Input: A string of characters (e.g., "HAT").

Output: Generate patterns by replacing characters with the numeric value 1 and process the patterns as described below:

1. **Replace one character at a time with 1:**
   - For each character in the string, replace it with 1, keeping the other characters unchanged.
   - Example for "HAT":
   - 1AT, H1T, HA1
2. **Replace one character at a time with 1:**
   - For each character in the string, replace it with 1, keeping the other characters unchanged.
   - E
     x
     a
3. **Replace two characters at a time with 1:**
   - Replace every combination of two characters with 1, keeping the remaining character unchanged.
   - If 1s are consecutive, replace them with their sum (e.g., 11T becomes 2T).
   - Example for "HAT":
     11T → 2T, H11 → H2, 1A1
4. **Replace all characters with 1:**
   - Replace all characters in the string with 1.
   - If there are consecutive 1s, sum them up (e.g., 111 becomes 3).
   - Example for "HAT": 111
     → 3

Final Output

For the string "HAT", the output should be: 1AT, H1T, HA1, 2T, H2, 1A1, 3.

4. Given a sorted array arr [] and a target value, the task is to count triplets (i, j, k) of valid indices, such that arr[i] + arr[j] + arr[k] = target and i < j < k.

Examples:

Input: arr[] = [-3, -1, -1, 0, 1, 2], target = -2

Output: 4

5. You are given an array prices[] where prices[i] represents the price of a given stock on day i. You want to maximize your profit by choosing a single day to buy one stock and choosing a different day in the future to sell that stock. Write a program to return the maximum profit you can achieve from this transaction. If no profit is possible, return 0.

6. Find **the "Kth" max and min element of an array:**

Given k, find the k-th smallest and k-th largest element in the array. Input: arr = [7, 10, 4, 3, 20, 15], k = 3

Output: Kth Smallest: 7, Kth Largest: 10

7. Sort a binary array with values 0, 1, and 2 using constant space and one pass (Dutch National Flag algorithm).

Input: [0, 2, 1, 2, 0]

Output: [0, 0, 1, 2, 2]

8. Find **longest consecutive subsequence:**

Return the length of the longest consecutive elements sequence. Input: [1, 9, 3, 10, 4, 20, 2]

Output: 4 (Sequence: 1, 2, 3, 4)

9. Given a number of bits and a number K. In one flip you can toggle exactly K consecutive bits. With only this flip operation available, convert the string into all 1. Input String: 0000110000 and K=3

Following are four flip operations by using which all bits converted into 1's.

Flip1-1110110000 Flip2- 1110110111

Flip3-1111000111 Flip4- 1111111111

If it is not possible to convert all bits into one's then print "IMPOSSIBLE".

10. Given a list of non-negative integers, arrange them in such a way that they form the largest possible number. Since the result can be very large, return it as a string in **O(N log N)** time complexity.

Example-1	**Example-2**
Input: N = 5 Arr[] = {3, 30, 34, 5, 9} **Output**: 9534330	**Input**: N = 4 Arr[] = {54, 546, 548, 60} **Output**: 6054854654

11. Given an array arr[] of size n containing distinct integers within the range [1, n+2], find the two missing numbers from the first n+2 natural numbers.

Constraints:

- The solution must run in **O(N) time** and use **O(1) extra space**.
- The array does not contain duplicate values.

Examples:

Input: arr[] = [1, 2, 4, 6, 3, 8], n = 6

Output: 5, 7

12. Given a string str of lowercase alphabets and a number k, the task is to print the minimum value of the string after removal of k characters. The value of a string is defined as the sum of squares of the count of each distinct character present in the string. Return the minimum possible required value. **Examples:**

Input: str = "abccc", k = 1

Output: 6

Input: str = “aabcbcbcabcc”, k = 3
Output: 27
Expected Time Complexity: O(n+klog(p))
Note: Here n is the length of string and p is number of distinct alphabets and k number of alphabets to be removed.

13. Given a non-negative integer **S** represented as a string, remove **K** digits from the number so that the new number is the smallest possible.

Note : The given *num* does not contain any leading zero.
Expected Time Complexity: O(|S|).

Example 1:	**Example 2:**
Input:	**Input**:
S = "149811", K = 3	S = "1002991", K = 3
Output:	**Output:**
111	21

14. You are given a two-dimensional grid board[][] of size n * m consisting of English letters and a string target. Your task is to determine whether the target word can be formed by sequentially connecting letters from the grid. You may move to adjacent cells **horizontally or vertically** (not diagonally), and **a cell may not be reused** once it is part of the current path.

Examples:

Input:
board[][] = [['C', 'A', 'T'], ['R', 'A', 'K'], ['T', 'O', 'N']],
target = "CART" **Output:**
true **Explanation:**
You can trace the word "CART" through the path: C → A → R → T (moving horizontally and vertically, without repeating cells).

15. Given an encoded string **s**, the task is to decode it. The encoding rule is:

- **k[encodedString],** where the **encodedString** inside the square brackets is being repeated exactly **k** times. Note that **k** is guaranteed to be a positive integer, and encodedString contains only lowercase er

Note: The test cases are generated so that the length of the output string will never exceed 10^5.
Examples:
Input: s = "1[b]"
Output: "b"
Input: s = "3[b2[ca]]"
Output: "bcacabcacabcaca"

Competitive coding list will be shared with the students.

Total Hours: 30 hrs.